



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

“DESARROLLO DE UNA PRUEBA DE CONCEPTO DE UN OSB”

Almudena Ochoa Lainez

Tutor: José Javier Astrain

Cotutor: Diego Martínez Pomares

Pamplona, 23 de Febrero de 2011

RESUMEN

Durante los años 2009 y 2010 la Dirección General de Modernización y Administración Electrónica (DGMAE) del Gobierno de Navarra ha realizado diversos estudios para determinar la conveniencia de adoptar un modelo de arquitectura basado en estándares SOA (*Service-Oriented Architecture*) que aporte beneficios y mejoras en la eficiencia y calidad de servicio, y permita aumentar la capacidad de innovación y la flexibilidad para incorporar los cambios que el Gobierno de Navarra pueda demandar en el futuro.

Se ha elegido a Oracle como socio tecnológico para la estrategia SOA cuyas soluciones de más interés son: *Business Process Analysis Suite*, *SOA Suite* y *WebLogic Suite*.

Para adquirir un mayor conocimiento de características y capacidades de las soluciones que permita la toma de decisiones, la Oficina Técnica (OT) de la DGMAE ha propuesto un proyecto cuyo objetivo es la realización de una prueba de concepto de uno de los elementos claves de dicha estrategia SOA como es el Bus de Integración o *Enterprise Service Bus* (en este caso *Oracle Service Bus*, en adelante OSB).

Este documento presenta el desarrollo de una prueba de concepto para Gobierno de Navarra sobre la conveniencia y ventajas que puede aportar un OSB en las arquitecturas y aplicaciones existentes y futuras. El objetivo principal de este PFC es el desarrollo de esta prueba de concepto implementando los escenarios solicitados y generando un informe de resultados de cada una de las pruebas realizadas, que ayude a los grupos de arquitectura y estrategia de Gobierno de Navarra en la toma de decisiones.

La prueba de concepto se diseña como un conjunto de casos de uso donde los diferentes grupos de arquitectura de la DGMAE describen situaciones y características que consideran de interés. A estos grupos que determinan los requerimientos de negocio en la metodología de proyectos de Gobierno de Navarra se les conoce como Gestores de Producto y para la prueba de concepto se distinguen los siguientes: Grupo de Arquitectura Funcional (GAF), Grupo de Arquitectura de Desarrollo (GAD), Grupo de Arquitectura de Infraestructuras (GAI) y Sección de Procedimientos Electrónicos (SPE).

INDICE

INTRODUCCIÓN	5
DOCUMENTO VISIÓN.....	8
Introducción	9
Antecedentes.....	9
Objetivos del proyecto	11
Alcance del sistema	11
Identificación del alcance	11
Estudio de la situación actual	12
Valoración del estudio de la situación actual.....	12
Usuarios participantes.....	14
Estructura del proyecto	16
Producto final	16
Definición de la arquitectura tecnológica	16
Tecnologías	20
Definición del plan de acción.....	24
Horario de trabajo	24
Entregables	24
Definición del Equipo de Proyecto	25
Herramientas	26
Planificación temporal (Diagrama de Gantt).....	27
Análisis de riesgos y planes de contingencia.....	28
Identificación de riesgos.....	28
Actuación ante riesgos	28
Plan de contingencia	29
Replanificaciones	29
Horario de trabajo	30
Planificación final.....	31
Estrategia de gestión del proyecto.....	32
Definición del plan de mantenimiento.....	32
Comunicación	32
Fechas de entrega.....	32
Especificación de los requisitos	32
DOCUMENTO DE ANÁLISIS Y DISEÑO	37
Análisis.....	38
Análisis de requisitos	38
Modelo y Análisis de casos de uso	46
Diseño	75
Diagramas de secuencia	75

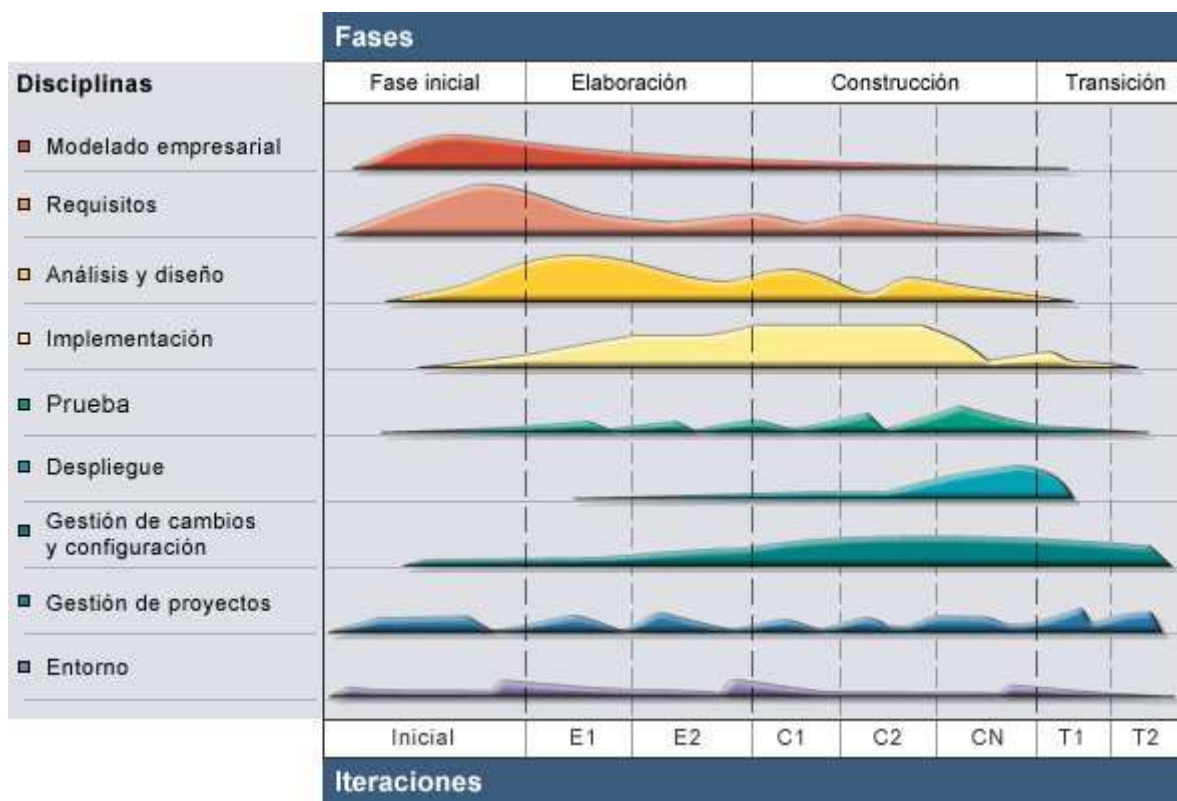
DOCUMENTO DE DESARROLLO / IMPLEMENTACIÓN.....	82
Implementación.....	83
Preparación del entorno de generación y construcción	83
Generación del código de los componentes	83
DOCUMENTO DE PRUEBAS Y VALIDACIÓN	86
Pruebas de validación.....	87
DOCUMENTO DE CONCLUSIONES	94
Presentación de informes a los usuarios finales	95
REFERENCIAS Y BIBLIOGRAFÍA.....	117
ANEXOS.....	119
Anexo I: ¿Qué tiene de diferente OSB 11.1.1.3.0 con respecto a las versiones anteriores?.....	120
Anexo II: Documentos de Implementación	123
CASOS DE USO DEL GAF.....	123
Caso de Uso 1: Publicar y proteger con WS-Security un WS sin protección.....	123
Caso de Uso 2: Escalabilidad en servicios síncronos	142
Caso de Uso 3: Escalabilidad en canal punto a punto	155
Caso de Uso 4: Transformación de mensajes.....	173
CASOS DE USO DE SPE	191
Caso de Uso 1: Consumidor de mensajes (Polling consumer)	191
Caso de Uso 2: Canal punto a punto confiable	201
Caso de Uso 3: Canal de publicación / suscripción	217
Caso de Uso 4: Encaminamiento basado en el contenido	233
Caso de Uso 5: Separador.....	241
Caso de Uso 6: Escucha en llamada síncrona	259
Caso de Uso 7: Canal de entrega garantizada	272
Caso de Uso 8: Canal de mensajes fallidos (dead letter channel).....	284
CASOS DE USO DEL GAD	290
Caso de Uso 1: “Invocación de WS con tipos de datos propietarios de .net desde Java	290
Anexo III: Entregables de diseño	305
Informes con <i>Oracle Service Bus</i>	305
Diseño proceso de gestión de cambios con <i>Oracle Service Bus</i>	314
Anexo IV: Glosario de términos.....	315

INTRODUCCIÓN

La metodología que se va a seguir en el proyecto es el proceso unificado de desarrollo de software (RUP). Se ha realizado esta elección ya que dicha metodología se adapta al contexto existente gracias a los ciclos iterativos, a su vez divididas en grandes fases o etapas por lo que hace más fácil su desarrollo.

El ciclo de vida RUP es una implementación del desarrollo en espiral. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 1 muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



-Figura 1. Ciclo de vida RUP-

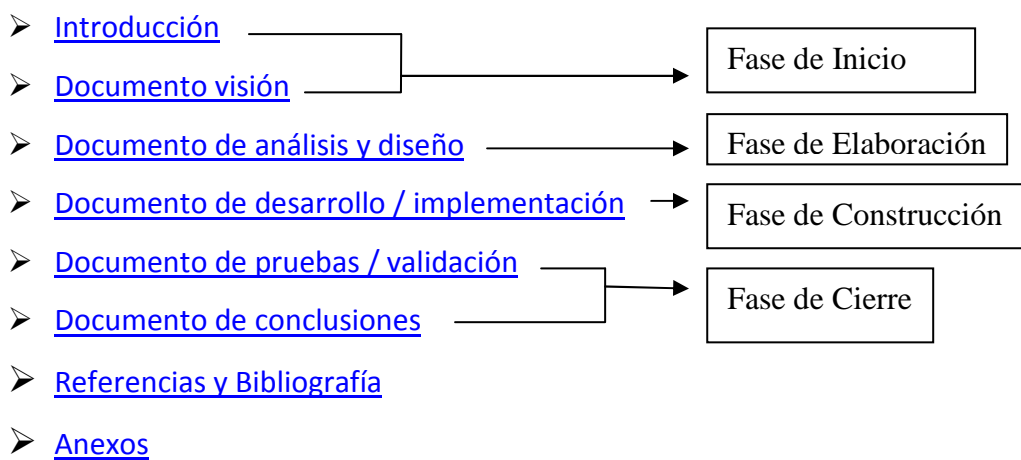
Fase de inicio: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Fase de cierre o transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

Siguiendo esta metodología, la memoria se divide en los siguientes capítulos:



DOCUMENTO VISIÓN

Introducción

El PFC que se va a desarrollar se realiza para el Servicio de Infraestructuras y Centro de Soporte de Gobierno de Navarra a través de PIN S.L. (Producción Informática de Navarra, S.L.) que es una empresa pública perteneciente al Gobierno de Navarra, que fue constituida en el año 2006 con la finalidad de proveer al mismo y a los organismos e instituciones que dependen de él, de sistemas de información, encargándose, igualmente, de la explotación de los mismos y teniendo la posibilidad de prestar todo tipo de servicios anexos.

El objeto de este proyecto es desarrollar una prueba de concepto, es decir, un desarrollo resumido e incompleto de una arquitectura de servicio (en este caso OSB), cuyo propósito es verificar que el concepto en cuestión es susceptible de ser empleado y conocer su viabilidad y consideraciones a tener en cuenta. Es el paso previo a la construcción de un prototipo operativo. En este sentido, en este proyecto no se va a desarrollar un producto final, sino que se van a desarrollar diferentes casos de prueba, generando un informe de resultados de cada una de ellas para extraer ventajas o desventajas de la utilización del OSB en cada uno de los escenarios planteados.

Los diferentes departamentos de Gobierno de Navarra exponen qué es lo que quieren probar, bajo qué circunstancias y qué resultados esperan obtener. Estas pruebas serán implementadas según los requerimientos planteados obteniendo como resultado un informe por cada una de ellas, indicando la viabilidad y eficiencia del uso del OSB en cada uno de los casos.

Se pretende que la prueba de concepto resuelva las siguientes incertidumbres actuales:

- Beneficios del bus de integración en ciertos escenarios y casos de uso.
- Conveniencia de uso, identificación de los casos en los que está recomendado su uso y en cuáles está desaconsejado.
- Determinar qué funcionalidades conviene utilizar y cuáles no aportan beneficio.
- Nivel de impacto en el ciclo de vida del desarrollo y mantenimiento de los proyectos y servicios.
- Impacto del bus de integración sobre los elementos ya existentes: conocer si existen requerimientos impuestos por el bus en los desarrollos.

Antecedentes

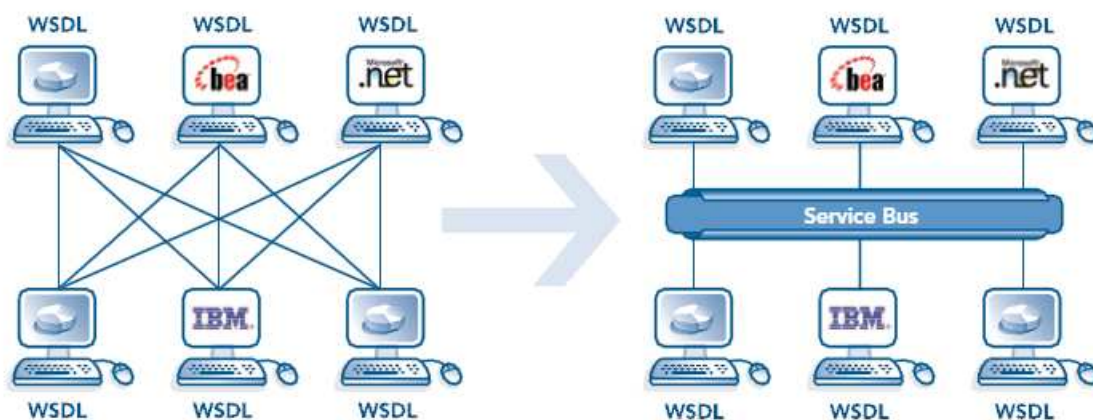
En cuanto a la empresa:

El Simple Object Access Protocol (SOAP, ver anexo IV), el lenguaje de descripción de Servicios Web (WSDL) y el Hypertext Transfer Protocol (http) han resuelto un problema

complejo que había fracasado a los intentos precedentes de crear estándares para sistemas distribuidos, como la *Common Object Request Broker Architecture* (CORBA) o el *Distributed Computing Environment* (DCE): la aceptación general de los dos campos tecnológicos de aplicación predominantes (.NET y J2EE) así como de un número de proveedores de aplicaciones. Aunque no debe subestimarse lo que esto supuso, lo cierto es que la aceptación se logró mediante la simplificación exhaustiva de los estándares, lo que permitió establecer un conjunto esencial y acordado de definiciones tecnológicas. Este principio de diseño es, al mismo tiempo, la mayor virtud y la mayor limitación del estándar de servicios Web.

En pocas palabras: ninguna implementación a escala empresarial puede funcionar exclusivamente con SOAP, WSDL y HTTP. Con su rápida aceptación entre la comunidad de la tecnología de la información (TI), los Servicios Web y la SOA se enfrentan a los mismos retos a los que se han enfrentado las generaciones anteriores de sistemas distribuidos, sólo que a mayor escala debido a sus principios de descomposición inherentes. El principal desafío es cómo resolver la escalabilidad de las conexiones punto a punto, lo que se conoce también como el “problema de conexión M*N”.

En la Figura 1, se muestra el problema de la explosión de conexiones que generan los enfoques de integración punto a punto. El número de conexiones crece exponencialmente por cada aplicación que se añade, a medida que cada aplicación se conecta a una nueva aplicación. Para ello se inventó el paradigma del *Enterprise service bus*, que se muestra a la derecha de la ilustración. Usando ese modelo, cada aplicación se conecta sólo una vez a una infraestructura troncal común: el bus.



-Figura 1 Paradigma del Enterprise Service Bus-

Esto reduce al mínimo el número de conexiones y proporciona una ubicación centralizada para su administración y para la gestión de sistemas integrados.

Para gestionar la complejidad referente a cómo un servicio cliente se conecta y se comunica con el proveedor del servicio, la SOA precisa una infraestructura troncal capaz de ir más allá de la mensajería distribuida tradicional para proveer transformación compleja, enrutamiento y conectividad acoplada libremente en un entorno TI heterogéneo, independientemente de las plataformas usadas. Esta infraestructura troncal fiable proporciona un bus de servicios a escala empresarial que responde verdaderamente, en toda su dimensión, a las siglas ESB.

En cuanto al producto:

En un principio, Oracle contaba con OESB (*Oracle Enterprise Service Bus*), que era un desarrollo de la propia compañía. No hay que confundir OESB con OSB.

Oracle Service Bus (OSB) es una versión renombrada de *AquaLogic Service Bus* (ALSB) de BEA Systems. ALSB fue liberado por BEA Systems en 2005. A mediados de 2008, Oracle Corporation adquirió el producto de BEA Systems. El producto ALSB fue renombrado a OSB.

La versión inicial de OSB es la versión 10.3 y la que se utiliza en este documento es la 11g. (Para más información consultar el Anexo I: ¿Qué tiene de diferente OSB 11.1.1.3.0 con respecto a las versiones anteriores?).

Objetivos del proyecto

Los objetivos del proyecto son:

- Conocer, mediante la realización de la prueba de concepto, las características, requerimientos, ventajas y limitaciones de un bus de integración en los futuros proyectos de Gobierno de Navarra.
- Obtener un informe de resultados de los diferentes casos de uso planteados que permita a los “solicitantes” conocer el resultado de la prueba y permita la futura toma de decisiones.

Alcance del sistema

Identificación del alcance

Dentro del alcance:

1. Toma de requerimientos para la prueba de concepto

- Casos de uso concretos a desarrollar con una descripción de la prueba a realizar, elementos a utilizar (a ser posible, elementos existentes en preproducción o, si se considera, en producción), resultados esperados y parámetros de medida.

Para dicha toma de requerimientos y casos de uso se va a contar con los siguientes grupos, pertenecientes a diferentes departamentos de Gobierno de Navarra (más información apartado [Usuario participantes](#)):

- Grupo de arquitectura de infraestructuras.
- Grupo de arquitectura funcional.

- Grupo de arquitectura de desarrollo.
- Sección de procedimientos electrónicos.

Inicialmente se va a proponer al Grupo de Arquitectura Funcional que exponga sus casos de uso y posteriormente se extenderá al resto de grupos, de este modo aquellos casos de uso ya cubierto por el GAF no serán necesario proponer por otros grupos (nota: se considera que el GAF expondrá como necesidades un conjunto bastante amplio y significativo de casos de uso).

2. Desarrollo de la prueba de concepto simulando los casos de uso propuestos y obteniendo resultados y valores de medida solicitados.
3. Obtención de los siguientes entregables:
 - Guía de referencia de la implementación de cada caso de prueba.
 - Informe final a alto nivel.
 - Casos desplegados (y desplegables) en el entorno de laboratorio.
4. Realización de una reunión de presentación de resultados.

Fuera del alcance:

Los elementos fuera del alcance del proyecto son:

- Implantación de la infraestructura del Bus de Integración en producción. Este punto será objeto de otro proyecto y recibirá como entrada los datos de la prueba de concepto.
- Formación a los usuarios.
- Paso a producción de cualesquiera de los resultados del proyecto.
- Extender la prueba de concepto a los grupos de arquitectura siguientes:
 - PIN, S.L.
 - Salud, a través del personal técnico del Servicio de Sistemas de Información del Área Sanitaria de Gobierno de Navarra.

Estudio de la situación actual

Valoración del estudio de la situación actual

Durante los años 2009 y 2010 la Dirección General de Modernización y Administración Electrónica ha realizado diversos estudios para determinar la conveniencia de adoptar un modelo de arquitectura basado en estándares SOA que aporte beneficios y mejoras en la eficiencia y calidad de servicio, así como aumentar la capacidad de innovación y

la flexibilidad para incorporar los cambios que el Gobierno de Navarra pueda demandar en el futuro.

A modo de ejemplo se citan los proyectos que apoyaron tal idea y mostraron el camino, tanto desde el punto de vista organizativo como tecnológico, para llegar a tal objetivo. Proyectos como el Libro Blanco Java de Gobierno de Navarra¹ o el *Oracle Insight* mostraron a la DGMAE los beneficios de ir adoptando un modelo basado en SOA.

La DGMAE se ha interesado en los siguientes productos de la estrategia SOA:

- *Business Process Analysis Suite*
- *SOA Suite*, que incluye los siguientes productos
 - *Oracle BPEL Process Manager*
 - *Oracle Service Bus*
 - *Oracle Web Services Manager*
 - *Oracle Business Activity Monitoring*
 - *Oracle Business Rules Engine*
 - *Oracle Complex Event Processing*
- *WebLogic Suite*, que incluye los siguientes productos
 - *WebLogic Server Enterprise Edition*
 - *Internet Application Server Enterprise Edition*
 - *Coherence Enterprise Edition*
 - *WebLogic Operations Control*
 - *JRockit Real Time (100ms) y Jrockit Mission Control*
 - *Application Diagnostics for Java (AD4J)*
 - *Application Development Framework*

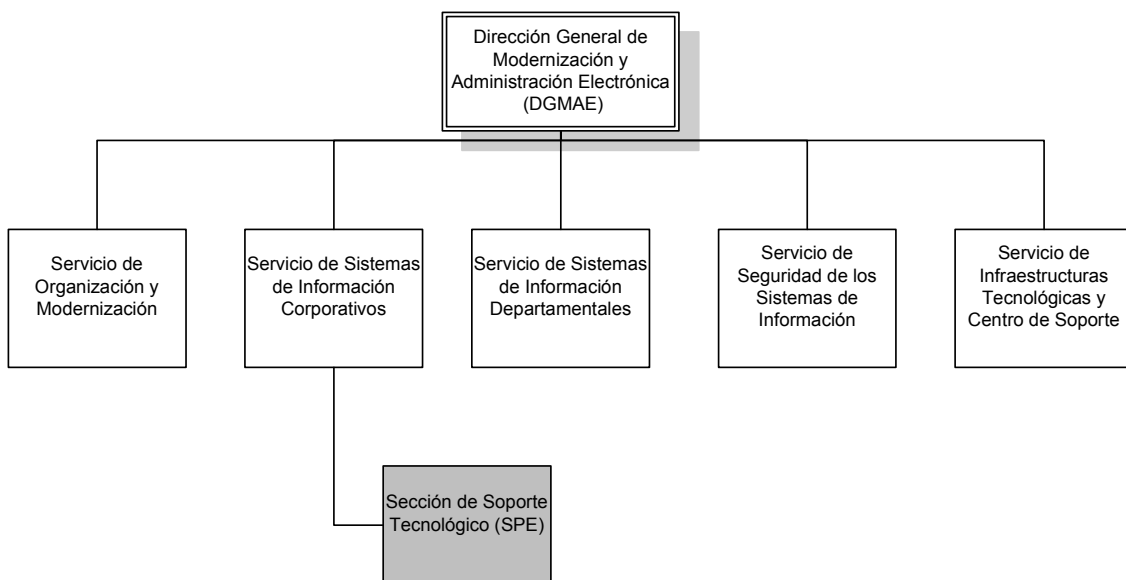
Una vez elegida la tecnología se aborda este nuevo proyecto para realizar una prueba de concepto de uno de los elementos claves de dicha estrategia SOA como es el Bus de Integración o *Enterprise Service Bus*. El objetivo de la prueba de concepto es instalar un laboratorio para realizar pruebas definidas que representen posibles casos de uso del Bus de Integración en las futuras soluciones a implantar en Gobierno de Navarra así como determinar la conveniencia o no de utilizarlo en ciertos casos. También se quiere determinar con exactitud los beneficios que se obtienen así como las implicaciones, tanto desde el punto de vista de rendimiento como de cambios en el ciclo de vida del desarrollo y mantenimiento, de un producto y tecnología de éste tipo.

¹Referencias a dichos proyectos son confidenciales por Gobierno de Navarra.

En resumen, la prueba de concepto pretende implantar un laboratorio para realizar pruebas de concepto de un Bus SOA, como una de las piezas clave de la estrategia SOA de esta Dirección General. El producto sobre el que se realiza la prueba de concepto SOA es *Oracle Service Bus* (OSB).

Usuarios participantes

En la Figura 2 se presenta la estructura principal de la DGMAE. En relación a este proyecto cabe destacar el grupo SPE (Sección de Soporte Tecnológico) que es uno de los participantes de los casos de prueba. El principal objetivo de este grupo es la gestión de todos los estándares tecnológicos de información y aplicaciones de la administración.

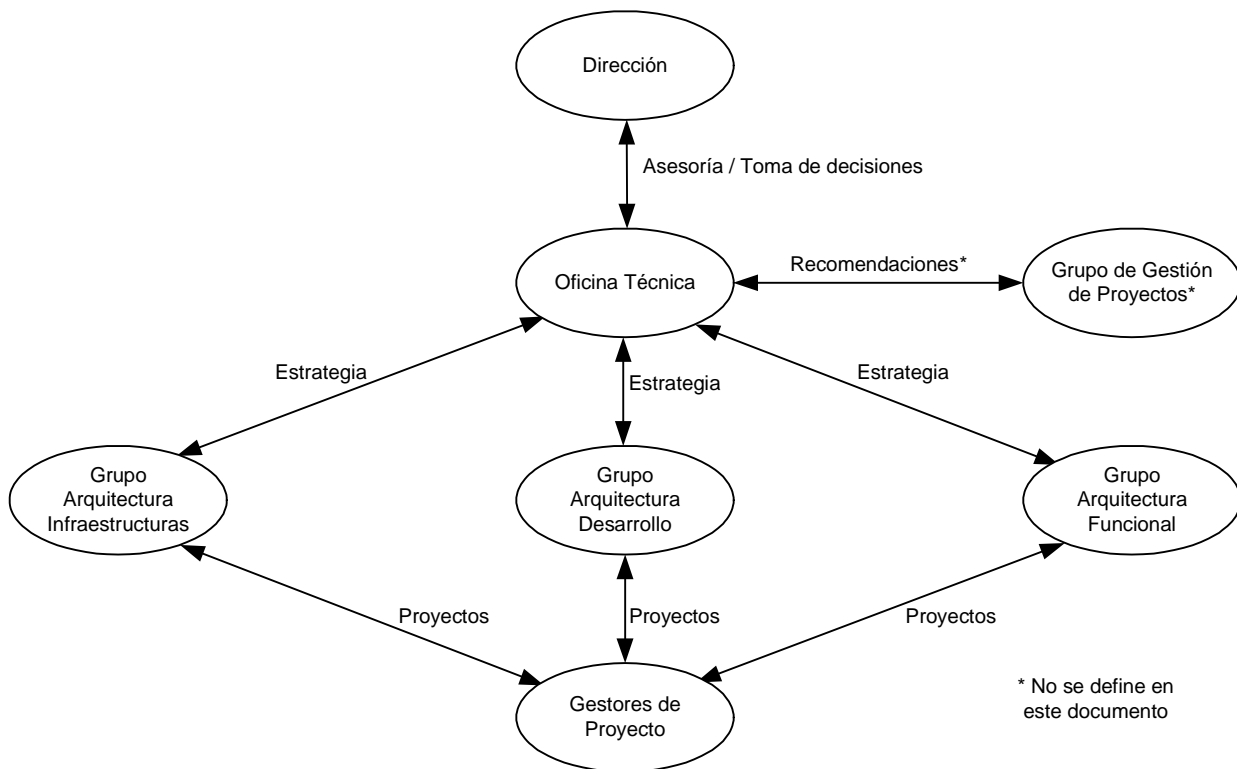


-Figura 2. Estructura de la DGMAE-

La DGMAE tiene una Oficina Técnica (OT) cuyos principales objetivos son proporcionar ayuda estratégica a la dirección, implantar una cultura de reutilización y facilitar la visión del escenario tecnológico a los equipos de proyecto.

La oficina técnica coordina tres grupos de arquitectura: grupo de arquitectura de infraestructuras, grupo de arquitectura de desarrollo y grupo de arquitectura funcional. Los grupos de arquitectura velan porque los proyectos se adecuen al escenario tecnológico de la DGMAE, en concreto, el Grupo de arquitectura de infraestructuras vela por la estandarización de la arquitectura de ejecución, el Grupo de arquitectura de desarrollo vela por la estandarización de las herramientas y plataformas de desarrollo y el Grupo de arquitectura funcional vela por la estandarización de los componentes software reutilizables.

La Figura 3 ilustra las relaciones existentes entre los diferentes cargos y grupos con relación a la oficina técnica de la DGMAE.



-Figura 3.Relaciones entre perfiles de usuario-

Por lo tanto, los principales beneficiarios directos son:

- Grupo de Arquitectura Funcional, que obtendrá información sobre la posibilidad y beneficios de utilizar el bus de integración como elemento de publicación de todos, o algunos, de los elementos reutilizables, así como las capacidades de integración del bus con la herramienta del Catálogo de Elementos Reutilizables.
- Grupo de Arquitectura de Desarrollo y Arquitectos de diferentes áreas de Gobierno de Navarra, que podrán evaluar casos de uso para conocer la viabilidad técnica y conveniencia de incorporar el bus de integración en sus proyectos actuales o futuros
- Desarrolladores, que contarán con ejemplos de implementación, y se beneficiarán de problemas anteriores resueltos, etc.
- Personal de infraestructuras (Grupo de Arquitectura de Infraestructuras), que tendrán información sobre los beneficios que pueden obtener con un bus de integración, información que deberá ser el *input* del futuro proyecto de implantación del bus.
- Grupo de sección de soporte tecnológico.

Los beneficiarios indirectos, por supuesto, son los ciudadanos, ya que la implantación de una arquitectura SOA redundará en la optimización de los recursos de la DGMAE.

Estructura del proyecto

Para continuar detallando la forma en que se realizará el proyecto, se propone dividirlo en las siguientes fases:

- Fase I: Visión y alcance.
- Fase II: Desarrollo de la prueba de concepto.
- Fases III: Presentación de Resultados y Conclusiones.

Producto final

El producto final se puede dividir en dos partes:

- Entregables de diseño: se trata de un entregable, habitualmente un documento, orientado a conocer con mayor detalle funcionalidades del producto y pensar en su integración con los entornos y procesos de Gobierno de Navarra.
- Casos de prueba: realización de una prueba que pruebe una funcionalidad en concreto, dando un caso de prueba definido y donde se indica claramente qué valores de entrada se van a utilizar y qué resultados se esperan. Se utilizarán en la medida de lo posibles elementos ya existentes y sólo en caso de que no existan elementos reutilizables significativos se desarrollarán para validar su funcionamiento. Se deberán entregar informes con los resultados y conclusiones obtenidas por cada caso de uso.

Definición de la arquitectura tecnológica²

La arquitectura orientada a servicios (SOA), desde el punto de vista tecnológico, es el resultado de la constante evolución hacia un mayor desacoplamiento de las capas de una aplicación (presentación, orquestación de procesos y servicios de negocio) y a un mayor nivel de estandarización/interoperabilidad de cada una de estas capas.

Con un total desacoplamiento de las capas de las aplicaciones y un muy alto nivel de interoperabilidad, los beneficios desde el punto de vista tecnológico son claros:

- Favorece la reutilización y la reducción del *time to market*:
 - Aumenta el grado de reutilización al desacoplar las capas de una aplicación.
 - Permite reutilizar las aplicaciones existentes mediante la encapsulación en servicios.
 - Permite la utilización de servicios de terceros.
 - Permite reaprovechar las plataformas existentes.

² Información extraída de un estudio publicado por el Centro de Alto Rendimiento de Accenture (CAR) y de la documentación oficial de Oracle

- Aumenta la flexibilidad:
 - Simplifica la adaptación de los sistemas existentes.
 - Evita el desarrollo de interfaces punto a punto entre los sistemas.
 - Aumenta la interoperabilidad entre sistemas, permitiendo tanto la externalización como la prestación de servicios.
- Mejora la productividad de los procesos:
 - Aumenta el nivel de automatización de los procesos, reduciendo el número de actividades manuales.
 - Permite monitorizar la actividad del negocio (cuadros de mando).
 - Permite realizar un análisis estadístico de los flujos de negocio reales en base a indicadores clave de negocio, permitiendo la identificación de puntos de mejora a optimizar.
 - Permite evaluar el impacto y beneficio de variantes en los procesos mediante simulación.
- Mejora el proceso de construcción de software:
 - Favorece la implantación operativa.
 - Mejora la especificación de los requerimientos de negocio.
 - Proporciona una filosofía de desarrollo común a todos los negocios y canales.
 - Mejora la calidad del software.
 - Desacopla el desarrollo de servicios y de procesos.
 - Mejora el mantenimiento (procesos autodocumentados).
- Mejora la usabilidad de las aplicaciones:
 - Permite presentar al usuario la información dispersa en distintos sistemas y de forma integrada.
 - Permite alcanzar un mayor nivel de automatismo en las aplicaciones en procesos complejos de *workflow*.
 - Permite utilizar tecnologías de presentación avanzadas como Web 2.0.

Existen cinco factores importantes que aumentan el interés del equipo ejecutivo y sobre todo, de los responsables de desarrollo, por la arquitectura SOA:

1. La arquitectura SOA ayuda a mejorar la agilidad y flexibilidad de las organizaciones.
2. La arquitectura SOA permite una “personalización masiva” de las tecnologías de la información.
3. La arquitectura SOA permite la simplificación del desarrollo de soluciones mediante la utilización de estándares de la industria y capacidades comunes de industrialización.
4. La arquitectura SOA permite aislar mejor a los sistemas frente a los cambios generados por otras partes de la organización (protección de las inversiones realizadas).

5. La arquitectura SOA permite alinear y acercar las áreas de tecnología y negocio.

Existen 6 facilitadores tecnológicos principales que permiten, desde el punto de vista tecnológico, la implantación de SOA. No es imprescindible el uso de todos los facilitadores, pero cada uno de ellos es importante para alcanzar plenamente todos los beneficios esperados.

1. *Business Process Management* o BPM: es el principal facilitador que recoge las tecnologías para permitir la definición/ejecución de los procesos tanto en su componente de modelado (BPA) como de implantación (BPO). Adicionalmente, es aconsejable contar con un motor de reglas (BRE) que permita externalizar las reglas de decisión que requieren ser modificadas frecuentemente.

2. La tecnología de *Web Services*: permite encapsular los servicios mediante un estándar ampliamente aceptado por todos los fabricantes y proveedores. Este estándar proporciona ventajas claras para proveer y consumir servicios al exterior, pero no es obligatoria su implementación en entornos cerrados.

3. *Enterprise Service Bus* o ESB: facilita la conexión entre sistemas/servicios heterogéneos, resolviendo deficiencias de la tecnología de *Web Services* como la garantía de entrega, localización, seguridad, transaccionalidad, etc. Dependiendo de la heterogeneidad de una instalación, su uso puede ser imprescindible o no ser requerido.

4. *Business Activity Monitoring* o BAM: proporciona una monitorización de los procesos (con una visión de negocio) en tiempo real y con capacidad de actuación.

5. El Gobierno de desarrollo: *Enterprise Service Repositorio* o ESR, es el catálogo de servicios y procesos (tanto desde el punto de vista técnico como de negocio) y es fundamental para la gestión de los servicios y procesos tanto desarrollados como comprados.

6. El Gobierno de ejecución: es un conjunto de herramientas y utilidades que permiten el gobierno de los servicios y procesos en ejecución, generando cuadros de mando de niveles de servicio y aplicando políticas de actuación automáticas. Bajo este facilitador también se suelen cubrir todos los aspectos de seguridad en SOA.

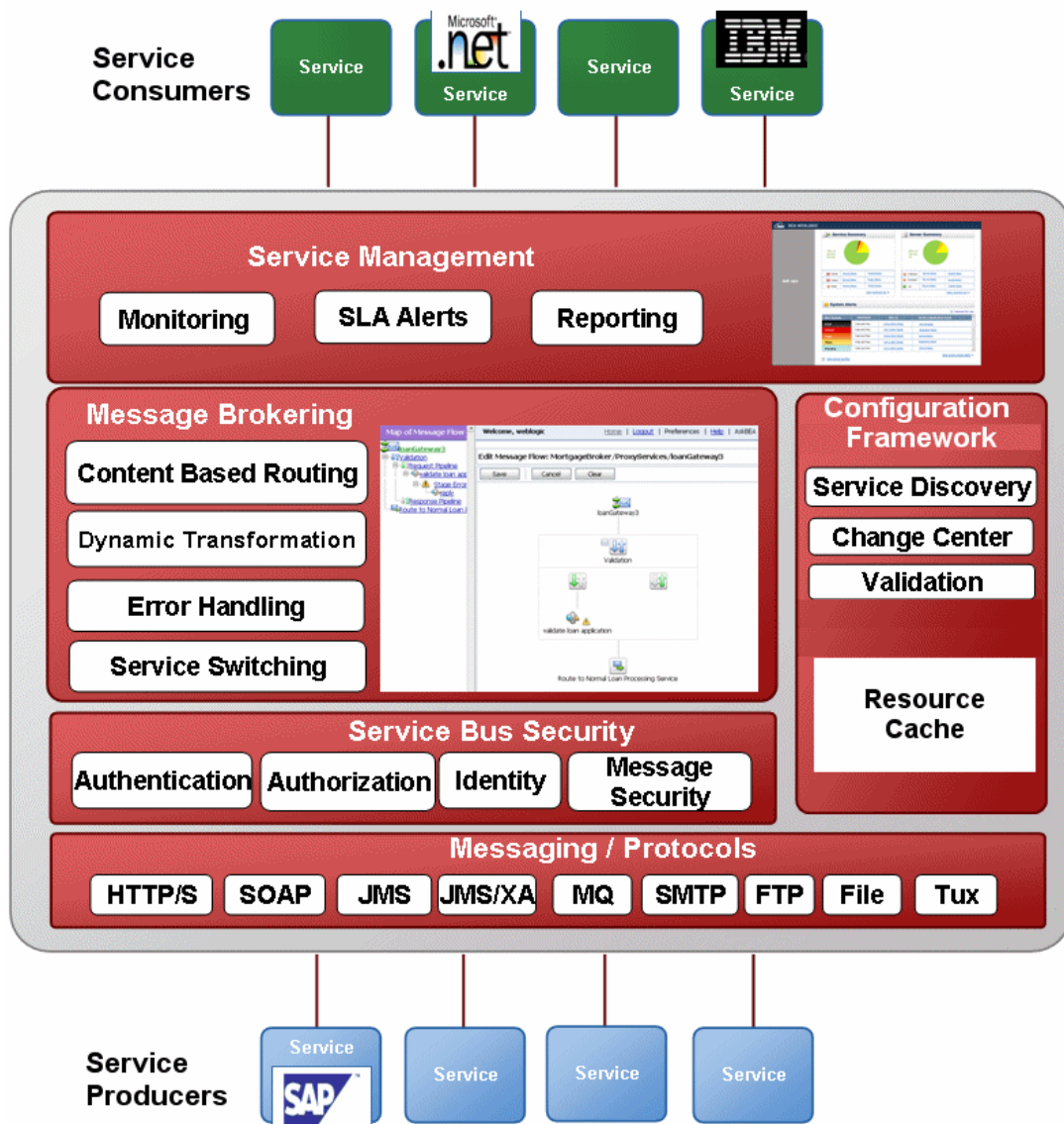
La arquitectura del *Oracle Service Bus* se centra alrededor de un Bus de Servicios Empresarial. El Bus ofrece servicios de entrega de mensajes, basados en estándares como SOAP, HTTP y *Java Messaging Service* (JMS). Diseñado para alto rendimiento, la entrega de mensajes garantizada a una variedad de productores y consumidores de servicios. Soporta XML como un tipo de datos nativo, mientras que también ofrece alternativas para el manejo de otros tipos de datos.

OSB es la política impulsada y le permite establecer un bajo acoplamiento entre los clientes de servicios y servicios de negocio, manteniendo al mismo tiempo un punto centralizado de control de seguridad y monitorización.

OSB es un intermediario que procesa los mensajes entrantes de solicitud de servicio, determina la lógica de enrutamiento, y transforma estos mensajes para la

compatibilidad con los consumidores de servicios. Recibe mensajes a través de un protocolo de transporte como HTTP (S), JMS, File y FTP, y envía mensajes a través de la misma o de un protocolo de transporte diferente. Los mensajes de respuesta siguen el camino inverso. El procesamiento de mensajes por parte de *Oracle Service Bus* es conducido por los metadatos, especificado en la definición del flujo de mensajes de un servicio de proxy.

La Figura 4 representa la arquitectura a alto nivel de *Oracle Service Bus* y sus subsistemas funcionales: gestión de servicios, la intermediación de mensajes, el marco de la configuración, la seguridad y la capa de transporte, y los protocolos de mensajería.



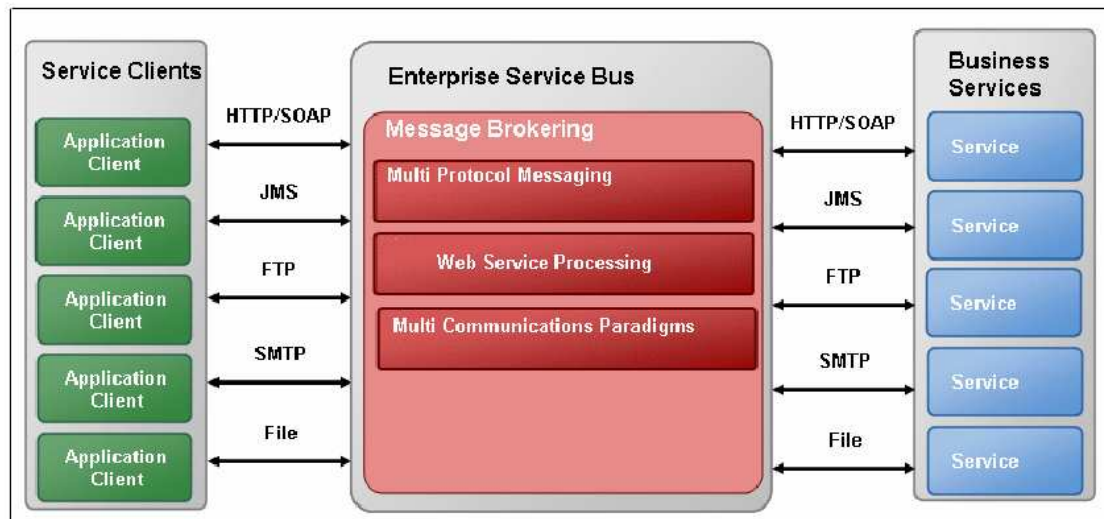
-Figura 4.Arquitectura a alto nivel de *Oracle Service Bus*

(http://download.oracle.com/docs/cd/E14571_01/doc.1111/e15020.pdf)-

La configuración de servicios sobre OSB se basa en la creación de un elemento de configuración de transporte (*business service*) y de un elemento de interacción (*Proxy Service*). La utilización de ambos elementos proporciona transparencia respecto a la

localización de los servicios, permitiendo minimizar el impacto de la sustitución de los mismos.

La Figura 5 ilustra la intermediación de mensajes por parte del OSB, es decir, permite el intercambio de mensajes entre los proveedores de servicios y un cliente a través de un Proxy de servicios y no directamente entre ellos, eliminando las complejidades derivadas de la heterogeneidad de los protocolos de comunicación y del formato de los mensajes.



-Figura 5. Intermediación de mensajes (Message Brokering) en OSB
(http://download.oracle.com/docs/cd/E14571_01/doc.1111/e15020.pdf)-

Tecnologías

Durante todo el documento se verán diferentes tecnologías (protocolos de comunicación) utilizados en todas las pruebas que se han hecho sobre el OSB. En esta sección se nombran los más importantes para tener una idea de los que puede llegar a tener un OSB.

FTP: el significado de las siglas es Protocolo de Transferencia de Archivos (*File Transfer Protocol*). Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El Servicio FTP es ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el *login* y *password* del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar ese tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

En el caso del OSB suele utilizarse como protocolo de entrada o salida de archivos. De esta forma los usuarios autorizados pueden dejar archivos (mensajes) que el OSB recogerá una vez detectados.

JMS: el significado de las siglas es Servicios de Mensajería de Java (*Java Message Service*). Permite que los componentes de aplicaciones hechos con JAVA, envíen, reciban, lean y escriban mensajes. Esto puede suceder tanto de manera síncrona como de manera asíncrona. Este servicio de mensajería también es conocido como Middleware Orientado a Mensajes (MOM).

Hay dos modelos de servicios de mensajería:

- El modelo punto a punto: en este caso un cliente envía un mensaje a otro cliente, si el receptor del mensaje no se encuentra activo el mensaje se encola hasta que lo esté.
- El modelo Publicador/Subscriber: en este otro caso tenemos un cliente que publica un tema o evento y (normalmente) más de uno que se subscriben a ese mensaje.

SMTP: El significado de las siglas es Protocolo Simple de Transferencia de Correo (*Simple Mail Transfer Protocol*). Se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1000 caracteres, aunque puede ampliarse usando las extensiones del servicio SMTP.

Las respuestas del servidor constan de un código numérico de tres dígitos, seguido de un texto explicativo. El número va dirigido a un procesamiento automático de la respuesta por autómata, mientras que el texto permite que un humano interprete la respuesta. En el protocolo SMTP todas las órdenes, réplicas o datos son líneas de texto, delimitadas por el carácter <CRLF>. Todas las réplicas tienen un código numérico al comienzo de la línea.

En la pila de protocolos TCP/IP, el SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

El OSB puede ser utilizado por el administrador como una fuente de *reports*. Cada vez que ocurra un error categorizado muy grave en un flujo, el propio OSB le enviará un *mail* de aviso.

HTTP: El significado de las siglas es Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*). Es el protocolo usado en cada transacción de la Web. HTTP fue desarrollado por el consorcio WEC y la IETF, colaboración que culminó en 1999 con la publicación de una serie de RFCs, siendo el más importante de ellos el RFC 2616, que especifica la versión 1.1.

HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, *proxies*) para comunicarse. Es un protocolo

orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador o un spider) se lo conoce como “*user agent*” (agente del usuario). A la información transmitida se le llama recurso y se le identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las *cookies*, que contienen información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de “sesión”, y también permite rastrear usuarios ya que las *cookies* pueden guardarse en el cliente por tiempo indeterminado.

Dentro de este protocolo también hace falta hacer mención de la conexión HTTPS, que se basa en el http añadiéndole seguridad, para ello utiliza un cifrado basado en TSL/SSL (*Transport Layer Security/Secure Sockets Layer*) para crear un canal seguro.

En el OSB es muy habitual encontrar este tipo de conexiones. Actualmente la mayoría de servicios en las empresas se ofrecen a través de *Web Services* tanto por HTTP como por HTTPS. Por lo tanto se suele usar generalmente como punto de destino.

SSL: proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas, la falsificación de la identidad del remitente y mantener la integridad del mensaje.

Como ejemplos más claros de protocolos que usan SSL están el HTTPS o el SFTP. Es interesante que los ESB también tengan acceso por protocolos que utilicen SSL porque actualmente en las empresas casi todo requiere seguridad de acceso.

EJB: Los EJB (*Enterprise Java Beans*) son una de las API que forman parte del estándar de construcción de aplicaciones de la plataforma J2EE y cuya especificación describe cómo proveer funcionalidad de objetos del lado del servidor con capacidades de control de transacciones, seguridad, concurrencia y eventos basados en JMS, entre otras. En este contexto, resultan una solución tecnológica muy importante para implementar y desplegar servicios y su composición, cuando se utiliza la variante de EJB de sesión. El otro enfoque de uso de EJBs lo abordan los EJB de entidad que encapsulan objetos del dominio como lo son las entidades del modelo entidad-relación.

Transporte FILE: En este caso se habla de transporte porque no es un protocolo propiamente dicho. Se está hablando de la transferencia de archivos en un entorno local. OSB suele utilizar este servicio para ir montando una estructura interna de

directorios para los diferentes servicios que no tienen que ser visibles por los demás usuarios.

En la Tabla 1 se hace un resumen de los protocolos de transporte que son usados por el OSB:

TIPO DE SERVICIO	PROTOCOLO DE TRANSPORTE
Servicio Web WSDL	HTTP(S), JCA, JMS , Local, SB, SOA-DIRECT, WS Petición JMS y respuesta JMS no están soportadas si <i>WS-Security</i> está activada.
Cualquier servicio SOAP (no WSDL)	HTTP(S), JMS, Local, SB Petición JMS y respuesta JMS no están soportadas si <i>WS-Security</i> está activada.
Cualquier servicio XML (no WSDL)	Email, File, FTP, HTTP(S), JMS, Local, MQ, SB, SFTP, Tuxedo HTTP GET sólo está soportado con XML con no WSDL.
Servicio de mensajes (Binario, Texto, MFL, XML)	Email, File, FTP, HTTP(S), JMS, Local, MQ, SB, SFTP, Tuxedo HTTP GET sólo está soportado con XML con no WSDL.
Servicio de transporte	EJB, Flow(Split-Join), JEJB

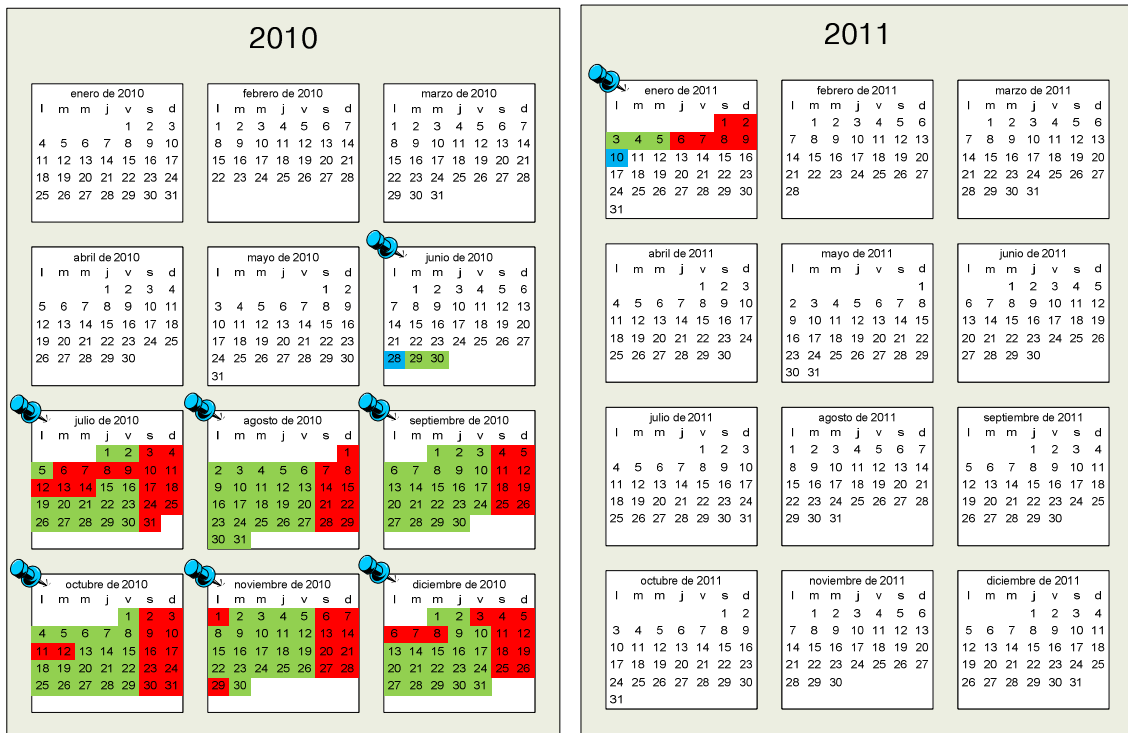
-Tabla 1.Protocolos de transporte-

Definición del plan de acción

Horario de trabajo



HORARIO: DE 8 A 15



- 26 Día Lectivo
- 25 Día Festivo
- 28 Principio y Fin del proyecto

Entregables

Al finalizar la prueba de concepto se deben entregar:

- Entregables Gobierno de Navarra:
 - Informes con los resultados obtenidos.
 - Implementación de las pruebas (proyectos generados).
 - Presentación final.
- Entregables Universidad Pública de Navarra:
 - Memoria: documento generado durante el proyecto que contiene: identificación de requisitos, análisis del sistema, diseño, etc. En esta memoria se incluyen los informes presentados a Gobierno de Navarra.

- Implementación de las pruebas (proyectos generados).
- Presentación: documento de la presentación final del proyecto.

Definición del Equipo de Proyecto

Empresa	Rol	Nombre	Responsabilidad
Gobierno de Navarra	Responsable del Proyecto	Oficina Técnica	
Gobierno de Navarra	Gestor de proyecto	Miembro del Grupo de Arquitectura de Infraestructuras	Asegurar que se alcanzan los objetivos del proyecto. Conduce las actividades del proyecto para cumplir con los objetivos dentro de las restricciones.
Gobierno de Navarra	Otros Gestores de Producto	Grupo de Arquitectura Funcional Sección de Procedimientos Electrónicos Grupo de Arquitectura de Desarrollo Grupo de Arquitectura de Infraestructuras	Definir requerimientos funcionales de la prueba concepto. Deben tener en mente posibles proyectos y casos de uso donde ven posible e interesante la utilización del bus con el objetivo de conocer su viabilidad y resolver incertidumbres.
Gobierno de Navarra	Desarrollador	Almudena Ochoa	Desarrollo de la prueba de concepto.
Gobierno de Navarra	Gestor de Implantación	Especialista Web	
Oracle	Consultor Especialista en <i>Oracle Service Bus</i>	Soporte de Oracle	Dar soporte sobre el producto y ayudar en la consecución de la solución.

-Tabla 2.Equipo de proyecto-

Herramientas

En el siguiente apartado se hace referencia tanto al hardware como al software necesario para el desarrollo del presente proyecto.

En cuanto al hardware que se va a utilizar para el desarrollo del proyecto, es un ordenador prestado por la propia empresa.

El OSB 11.1.1.3.0 (diferencias con las versiones anteriores en Anexo I) se encuentra instalado en una máquina virtual Red Hat con 8GB de memoria RAM y 2 procesadores.

En cuanto al software utilizado para el desarrollo de las diferentes pruebas son necesarias las siguientes herramientas de desarrollo:

- Eclipse 3.5.2 + Oracle Enterprise Pack for Eclipse 11g R1 (OEPE).
- JDeveloper 11.1.1.3.0
- NetBeans 6.9.1
- OSB 11.1.1.3.0
- Visual Studio 2008

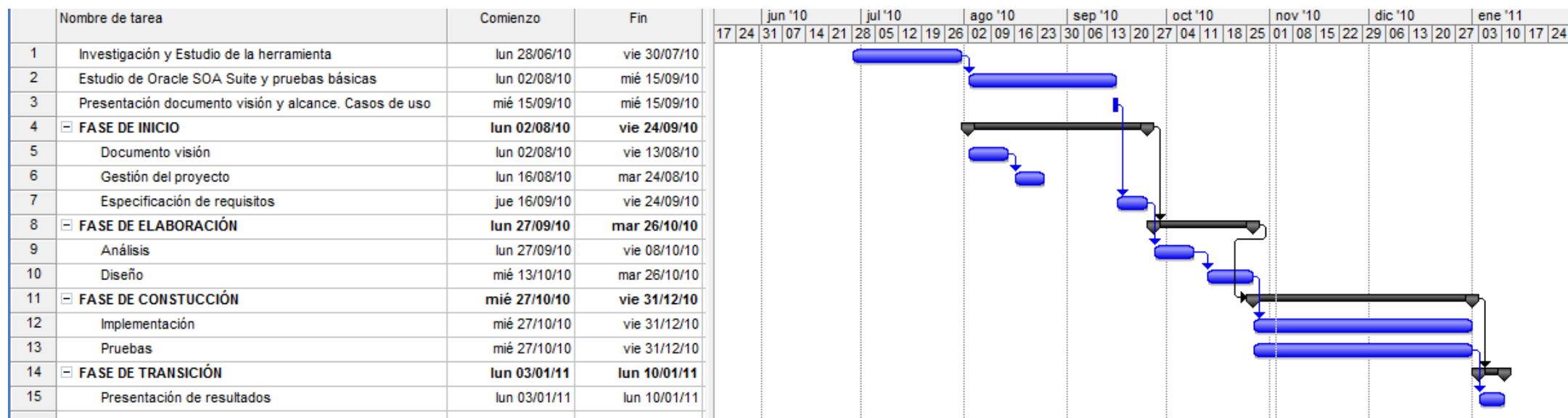
Tanto Eclipse como la consola del OSB se pueden utilizar para desarrollar los proyectos del bus, aunque en realidad uno es complementario del otro, es decir, en Eclipse se puede desarrollar el proyecto excepto realizar el debug (o test) y controlar la monitorización, que se realiza desde la consola del OSB. Por el contrario, en la consola del OSB se puede desarrollar un proyecto excepto crear transformaciones XQuery y Split-Join. Eclipse permite crear estas transformaciones de manera más cómoda a través de una interfaz gráfica.

Para poder utilizar Eclipse+OEPE es necesario instalarse el OSB en local aunque no es necesario ejecutarlo.

Es necesario utilizar JDeveloper para crear los adaptadores JCA (FTP, File, Base de datos...) aunque en estas pruebas no ha sido necesario utilizarlo con este fin, sino que se ha utilizado para crear los diferentes *Web Services*.

NetBeans y Visual Studio 2008 se utilizan para implementar las diferentes aplicaciones cliente.

Planificación temporal (Diagrama de Gantt)



Análisis de riesgos y planes de contingencia

Antes de comenzar con el desarrollo del proyecto es necesario identificar los posibles riesgos que se pueden presentar en el transcurso del mismo. Los siguientes riesgos que se presentan a continuación han sido definidos por el gestor del proyecto a la hora de definir el alcance del proyecto.

Identificación de riesgos

A lo largo de la duración del proyecto puede haber una serie de factores que influyan en su desarrollo:

- Se ha detectado que la documentación del fabricante es insuficiente, aunque se dispone de un libro sobre *Oracle Service Bus* ³(no documentación oficial del fabricante) con mayor detalle.
- Que la definición de los casos de uso no sea lo suficientemente detallada.
- Falta del único desarrollador.

Actuación ante riesgos

- Falta de recursos: si se produce la ausencia del encargado del proyecto por enfermedad o por cualquier otro motivo que impidan trabajar el número de horas establecidas por lo que no se cumple la planificación, se meterán horas extra en los días no destinados al desarrollo del proyecto para que aquellas actividades que se hayan visto perjudicadas no sufran retrasos.

Si no se dispone de los recursos necesarios, se debe informar al director del proyecto y buscar una alternativa. Si se produce un fallo del equipo, debemos tener una copia de seguridad para no perder nada y poder continuar en otro equipo.

- Mala planificación: si la duración del proyecto es mayor de la planificada se tendrá que negociar con el director del proyecto para reducir el alcance del mismo.
- Falta de documentación: si no se dispone de la documentación necesaria se dispone del soporte de Oracle para la resolución de los problemas que se puedan presentar.

³ The Definitive Guide to SOA: *Oracle Service Bus*, Second Edition. Jeff Davies, David Schorow, Samrat Ray y David Rieber. Apress 2008.

Plan de contingencia

Si se produce alguna de las opciones anteriores y retrasan la planificación del proyecto se deberá:

- Realizar una reunión con el director para analizar las causas del retraso.
- Estudiar las alternativas para superar dicho retraso.
- Planificar de nuevo las horas de trabajo incrementando el número de horas diarias e incluyendo horas extras.
- Si aun tomando medidas no se produce una disminución del retraso, se negociará con el director la reducción del alcance del proyecto.

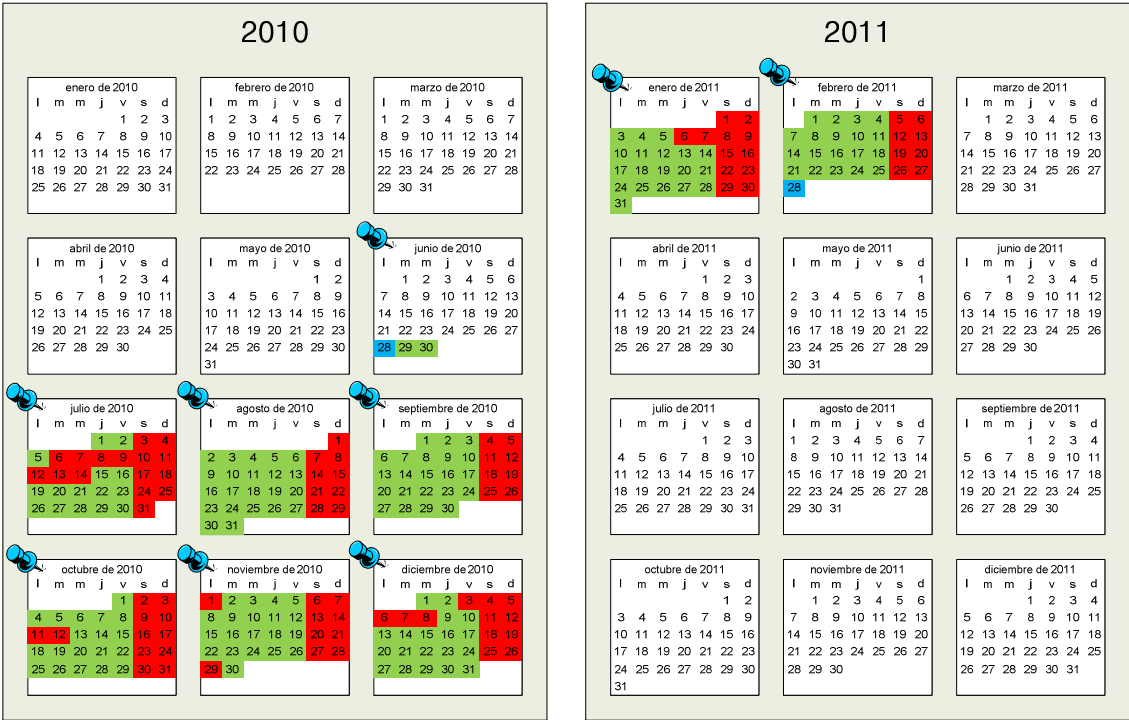
Replanificaciones

A continuación se presenta una modificación de la planificación inicial. Se ha retrasado dos meses aproximadamente la fecha de finalización del proyecto. La principal causa de este retraso ha sido que la fecha de entrega de los casos de prueba no se ha cumplido, es decir, los 'solicitantes' se han retrasado en la fecha de entrega, de Septiembre a Noviembre.

Horario de trabajo

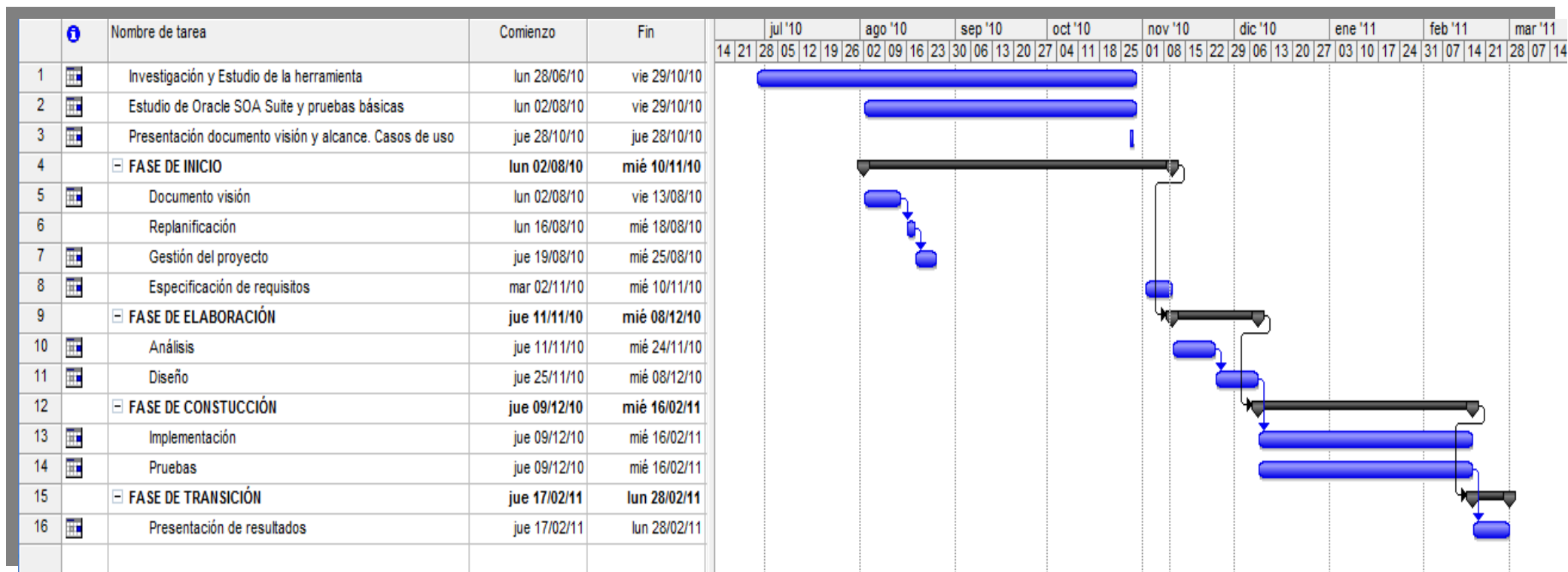


HORARIO: DE 8 A 15



- 26 Día Lectivo
- 25 Día Festivo
- 28 Principio y Fin del proyecto

Planificación final



Estrategia de gestión del proyecto

- Tiempo: Fijar (Fecha límite: 28 de Febrero de 2011)
- Recursos: Optimizar
- Funcionalidad: Aceptar

	Fijar	Optimizar	Aceptar
Fecha	✓		
Coste/Recursos			✓
Funcionalidad		✓	

Definición del plan de mantenimiento

Comunicación

Se realizará un seguimiento y control del proyecto de forma periódica por parte del responsable de la empresa Diego Martínez Pomares, además del seguimiento por parte del tutor del proyecto de la universidad José Javier Astrain Escola.

Se estará en comunicación con los 'solicitantes' ya sea a través de *emails* o reuniones para informales sobre el desarrollo del proyecto.

En cuanto al tribunal, se tiene una única reunión en la que el encargado del proyecto explica e informa sobre el desarrollo del proyecto.

Fechas de entrega

La duración estimada del proyecto es de ocho meses. Los cuatro primeros meses el desarrollo del proyecto será más lento ya que se necesita profundizar en el estudio de la herramienta. Los siguientes cuatro meses se destinarán principalmente al desarrollo de los diferentes casos.

Especificación de los requisitos

Un ESB o *Enterprise Service Bus* facilita la conexión entre sistemas/servicios heterogéneos, resolviendo deficiencias de la tecnología de *Web Services* como la garantía de entrega, localización, seguridad, transaccionalidad, etc. Dependiendo de la

heterogeneidad de una instalación, su uso puede ser imprescindible o no ser requerido.

Oracle Service Bus está diseñado para conectarse, mediar y administrar las interacciones entre servicios heterogéneos, aplicaciones tradicionales, aplicaciones empaquetadas y múltiples instancias *Enterprise Service Bus* (ESB) a través de una red de servicios para toda la empresa.

Oracle Service Bus permite la integración de servicios controlados por la configuración, con ruteo basado en identidades y contenido inteligente. Mejora la productividad del desarrollador debido a la integración de servicios de código libre.

Oracle Service Bus brinda capacidades incorporadas para la virtualización de servicios, *Web Service Security* (WS-Security) y el cumplimiento de políticas en torno a la regulación y la agrupación de servicios a fin de cumplir con los requerimientos de Confiabilidad, Disponibilidad y Desempeño (RASP) y evitar la sobrecarga de servicios de *back-end*. *Oracle Service Bus* está creado desde cero con soporte integral para SOA, Java 2 Platform, Enterprise Edition (J2EE), y estándares como *J2EE Connector architecture* (JCA), *WS-Reliable Messaging* y *WS-Security*.

Examinamos los requisitos a escala empresarial de un ESB. Cada una de las siguientes funciones es un elemento esencial para la satisfactoria integración de una SOA. Juntos, estos elementos resuelven los problemas a los que se enfrentan los clientes y los proveedores de servicios en un entorno de SOA.

- Mensajería distribuida. El núcleo del ESB lo constituye una aplicación de *middleware* orientada al mensaje. Este núcleo proporciona un método de transporte fiable y distribuido que emplea un mecanismo de almacenamiento y reenvío gracias al cual se garantiza la entrega de los mensajes incluso en caso de anomalías en la red.
- Transparencia de las ubicaciones. Con la mediación entre servicios, un servicio cliente que invoque al proveedor de servicio solo necesita saber que el servicio existe; el cliente no necesita saber dónde se está ejecutando el servicio. El ESB localiza el servicio cuando se invoca. Esto proporciona un cierto nivel de virtualización de los servicios y de transparencia de las ubicaciones, de forma que si un equipo falla, o si se cambia la ubicación de un proveedor de servicio, no es preciso notificar el cambio a cada uno de los clientes individuales. Esto puede contribuir significativamente a la reducción de los costes de gestión de las TI y a minimizar los riesgos.
- Transparencia del transporte. En los enfoques tradicionales de la integración punto a punto todos los componentes y objetos están muy estrechamente acoplados. En la SOA, los servicios están repartidos por todo el entorno de TI y su acoplamiento es menos estricto, gracias a la transparencia de las ubicaciones. Además de apoyarse en la transparencia de las ubicaciones para conectar clientes y proveedores de servicios, el ESB también proporciona protocolo de transporte físico para hacer posible la comunicación entre servicios utilizando transportes diferentes.

- Soporte multiprotocolo. Debido a que plantea cuestiones de fiabilidad inherentes y solamente funciona bien con patrones de intercambio de mensajes (MEP) síncrono, el modelo de transporte HTTP no satisface los requisitos de todos los servicios y aplicaciones. Por ejemplo, el servicio de mensajes de Java (JMS) además de poseer características asíncronas, ofrece más fiabilidad en el transporte que HTTP. Para compatibilizar el comportamiento de las aplicaciones individuales, algunos sistemas recurren a SOAP a través de JMS. También se usan otros tipos de modelos de transporte, entre los que se cuentan los sistemas de transporte propietarios de algunos de los principales proveedores de sistemas y soluciones de planificación de recursos empresariales. ESB necesita, por lo tanto, ser capaz de soportar muchos tipos de sistemas de transporte para integrar sistemas dispares y gestionar el transporte de comunicaciones complejas eficazmente.
- Calidad de servicio. En las aplicaciones empresariales, la calidad de servicio (QOS) hace referencia, fundamentalmente, a su fiabilidad. La entrega de los mensajes y la fiabilidad de servicios de invocación son funciones de misión crítica de cualquier sistema. Aún los servicios Web por sí solos no ofrecen un servicio de entrega garantizada. Un ESB, por otro lado, puede proporcionar un servicio de alta fiabilidad garantizando la entrega del mensaje de principio a fin que va más allá de la fiabilidad que puede ofrecer transportes como JMS. Asimismo, los métodos empleados para alcanzar un alto nivel de QoS deben satisfacer los estándares existentes, como, por ejemplo, ser compatibles con la especificación *WS-ReliableMessaging*⁴.
- Patrones de intercambio de mensajes. En la actualidad, la mayoría de los ESB se basan en un paradigma de solicitud/respuesta usando SOAP sobre HTTP; esto significa que el servicio cliente lanza un mensaje de solicitud al usuario y espera a recibir la respuesta. Esto se conoce como un MEP síncrono. Sin embargo, en el MEP de publicación/suscripción, el servicio cliente puede enviar un mensaje y suscribirse a la respuesta, en lugar de esperar a recibirla. El MEP de publicación/suscripción puede responder de forma más eficaz a eventos en un contexto empresarial, en particular cuando el ciclo de vida de una acción de servicio tiene lugar durante períodos de tiempo prolongados. Un ESB debe ser capaz de manejar ambos paradigmas.
- Enrutamiento basado en el contenido. Existen dos tipos de enrutamiento dentro de un ESB. El primer servicio de enrutamiento se produce cuando la invocación de un servicio entra en el ESB y éste encamina la respuesta al proveedor de servicio apropiado, sin necesidad de que el servicio cliente conozca la ubicación del proveedor del servicio. Así es cómo se logra la transparencia de las ubicaciones que antes hemos comentado. El otro tipo, enrutamiento basado en el contenido, introduce una serie de reglas o una lógica de negocio que se aplica al contenido del mensaje en la etapa del enrutamiento y que hacen posible que el ESB encamine los mensajes

⁴ Open Source SOA. Jeff Davis. Ed Manning 2009

a proveedores de servicio específicos basándose en su contenido; dando prioridad, por ejemplo, a los pedidos de determinados clientes o marcando los pedidos de gran tamaño para darles un tratamiento especial. Esto ofrece un servicio muy valioso, ya que puede contribuir a reducir el coste de la gestión de la información, garantiza que se respeten los acuerdos a nivel de servicio y permite a las empresas centrarse en actividades para mejorar la satisfacción de sus clientes.

- Transformación. Si bien la tarea de un ESB es dirigir mensajes de un servicio al siguiente, hay ocasiones en que el formato de los datos de un servicio no satisface los requisitos del siguiente servicio. Por ese motivo, el ESB debe ser capaz de transformar los datos de un formato a otro.

Es importante hacer un análisis de las necesidades de los usuarios para poder instalar todos los recursos que sean necesarios para que los resultados sean óptimos.

A continuación se presenta una lista con las principales necesidades del usuario final. Existen dos tipos de objetivos claramente diferenciables, por una parte el desarrollo de entregables de diseño, es decir, documentos orientados a conocer funcionalidades del producto y por otra parte el desarrollo de los propios casos de uso. Cada una de estas necesidades se prioriza ya que como se define en la matriz anterior se dispone de una fecha de fin fija y un único desarrollador. Se priorizan con tres categorías: 0-opcional, 1-deseable, 2- requerido

Entregables de diseño	Ciente	Entregable de diseño	Prioridad
	GAI	Informes con <i>Oracle Service Bus</i>	2
		Diseño proceso de gestión de cambios con <i>Oracle Service Bus</i>	2

Casos de uso	Ciente	Caso de Uso	Prioridad
	GAF	Publicar y proteger con <i>WS-Security</i> un WS sin protección	2
		Publicar y proteger con autenticación integrada un WS sin proteger	1
		Escalabilidad en servicios síncronos	1
		Escalabilidad en canal punto a punto	1
		Transformación de mensajes	1
		Despliegue de un proceso de negocio CES en el bus	0
	SPE	Consumidor de mensajes (<i>Polling consumer</i>)	2
		Canal punto a punto confiable	2
		Canal de publicación / suscripción	2
		Encaminamiento basado en el contenido	1
		Separador	0
		Escucha en llamada síncrona	1
		Canal de entrega garantizada	2
		Canal de mensajes fallidos (dead letter channel)	2
	GAD	Invocación de WS con tipos de datos propietarios de .Net desde Java	2

-Tabla 3.Priorización de los Casos de Uso-

DOCUMENTO DE ANÁLISIS Y DISEÑO

Análisis

Análisis de requisitos

En este apartado se va a describir de una manera más específica cuál es el objetivo y alcance de los requisitos expuestos en el apartado anterior.

Entregables de diseño del Grupo de Arquitectura de Infraestructuras

ENTREGABLE 1. INFORME DE CAPACIDADES DE INFORMES Y ALERTAS CON ORACLE SERVICE BUS

Se pretende conocer las capacidades del OSB para presentar informes de nivel de servicio. También se quiere conocer las capacidades de monitorización que da OSB, definición de umbrales y generación de alertas, así como envío de alertas a destinatarios.

Entre otras cosas se pretende poder obtener, y el informe debería dar una solución si la hubiera a:

- Tipos de informes que se pueden obtener.
- Valores de medición.
- Capacidades de monitorización y alertas.
- Recepción de alertas.
 - Protocolos (¿snmp?, correo electrónico, ¿otros?).

ENTREGABLE 2. PROPUESTA DE PROCESO DE GESTIÓN DE CAMBIOS CON ORACLE SERVICE BUS

El objetivo es estudiar los pasos necesarios para la incorporación del OSB en el proceso de gestión de cambio y despliegues actual en Gobierno de Navarra.

Debe estudiar y proponer un escenario de gestión de cambio de proyectos que utilicen el bus. Dar respuesta al tipo de paquetes “desplegables” que se pueden generar. El objetivo es conocer qué alternativas hay para que los desarrolladores incorporen el bus como parte de sus desarrollos y que deberían entregar así como qué paquetes de despliegue entregarían a los gestores de despliegue.

Casos de Prueba del Grupo de Arquitectura Funcional

CASO DE USO 1. PUBLICAR Y PROTEGER CON WS-SECURITY UN WS SIN PROTECCIÓN

Hacer que el Bus de Integración publique un servicio (*Proxy Service* en nomenclatura OSB) protegido con WS Security (*UserNameToken*). El servicio de back-end (*Business Service* en nomenclatura OSB) sería un *Web Service* abierto, sin protección.

Los usuarios utilizados para autenticarse en la llamada al *Web Service* deberán ser de Directorio Activo (cuentas de servicio).

Elaborar un proyecto Web simple tanto en .NET como en Java que invoque el *Web Service* publicado por el OSB.

CASO DE USO 2. PUBLICAR Y PROTEGER CON AUTENTICACIÓN INTEGRADA UN WS SIN PROTEGER

Hacer que el Bus de integración publique un servicio (*Proxy Service* en nomenclatura OSB) protegido con autenticación integrada (*NEGOTIATE/NTLM*). El servicio de *backend* (*Business Service* en nomenclatura OSB) sería un *Web Service* abierto, sin protección.

Elaborar un proyecto (Web o Desktop) en .NET que invoque el *Web Service* INTEGRADO publicado por el OSB.

CASO DE USO 3. ESCALABILIDAD EN SERVICIOS SINCRONOS

Alcance: El objetivo es demostrar la escalabilidad que proporciona contar con varios servicios de negocio iguales.

Paso 1

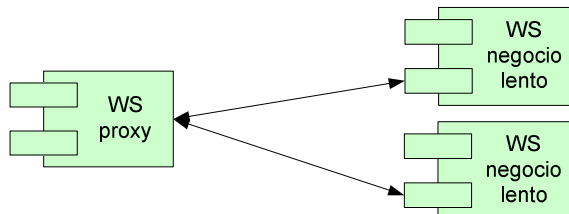
Tendremos un servicio Web SOAP de negocio que responderá de forma deliberadamente lenta.



Una aplicación cliente de prueba llamará al servicio proxy un número determinado de veces. Se tardará mucho tiempo en procesar todas las llamadas.

Paso 2

Ahora tendremos dos instancias del servicio de negocio lento.



La aplicación cliente llamará al proxy y se tardará, aproximadamente, la mitad de tiempo que antes en procesar todas las llamadas.

CASO DE USO 4. ESCALABILIDAD EN CANAL PUNTO A PUNTO

Alcance: El objetivo es demostrar la escalabilidad que proporciona un canal punto a punto con varios consumidores iguales.

Paso 1

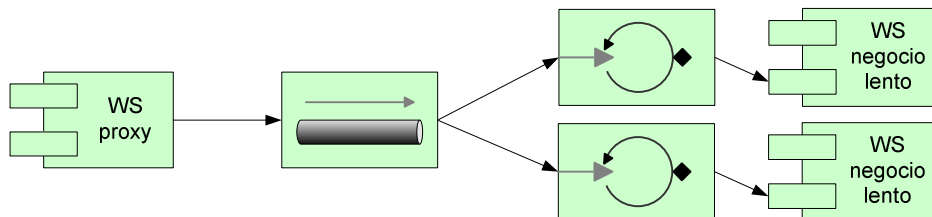
Tendremos un servicio Web SOAP de negocio de un solo sentido que consumirá los mensajes de forma deliberadamente lenta.



Una aplicación cliente de prueba llamará al servicio proxy un número determinado de veces. Se tardará mucho tiempo en procesar todas las llamadas.

Paso 2

Ahora tendremos dos instancias del servicio de negocio lento, y entre el proxy y los servicios un canal punto a punto.

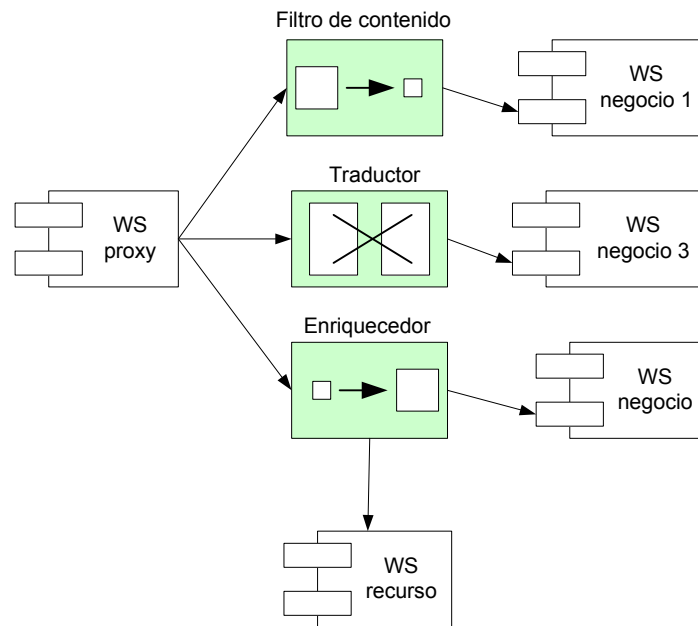


La aplicación cliente llamará al proxy. Los mensajes se entregarán casi inmediatamente, y se tardará, aproximadamente, la mitad de tiempo que antes en que todos los mensajes sean consumidos.

CASO DE USO 5. TRANSFORMACIÓN DE MENSAJES

Alcance: El objetivo es demostrar la transformación de mensajes que proporciona el bus.

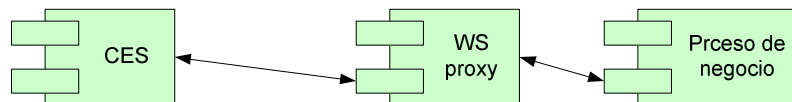
Tendremos tres servicios de negocio y un servicio proxy. El primer servicio de negocio espera un mensaje cuyo contenido sea parte del original. El segundo, espera un mensaje con el mismo contenido pero de manera diferente (otro formato, distinto número de campos). El tercero espera un mensaje cuyo contenido sea el original enriquecido por otro servicio auxiliar.



CASO DE USO 6. DESPLIEGUE DE UN PROCESO DE NEGOCIO CES EN EL BUS

Alcance: El objetivo es demostrar que se pueden desplegar procesos de negocio CES en el bus y CES los llame sin problemas, de manera que se puedan monitorizar como cualquier servicio desplegado en el bus.

Publicaremos el proceso de negocio (que es un servicio Web SOAP) en el bus y construiremos el correspondiente servicio Proxy. Configuraremos CES para que encamine los mensajes a ese servicio de la manera habitual.

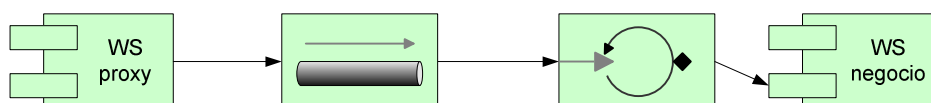


La aplicación cliente enviará mensajes a CES que los redirigirá al servicio de negocio publicado en el bus.

Casos de Prueba de la Sección de Procedimientos Electrónicos

CASO DE USO 1. CONSUMIDOR DE MENSAJES (POLLING CONSUMER)

Alcance: El objetivo es demostrar la implementación de un consumidor de mensajes que actúe de intermediario entre un servicio Web de un solo sentido y una cola de mensajes.



Tendremos un servicio de negocio SOAP síncrono de un solo sentido y una aplicación que llamará a un proxy que también será un servicio SOAP síncrono. Entre el proxy y el

servicio de negocio tendremos una cola de mensajes punto a punto y un consumidor que sacará los mensajes conforme lleguen y los reenviará al servicio de negocio. La aplicación de prueba enviará un determinado número de mensajes y todos llegarán al servicio de negocio.

CASO DE USO 2. CANAL PUNTO A PUNTO CONFIABLE

Alcance: El objetivo es demostrar la implementación de la integración basada en mensajes con un canal punto a punto y su tolerancia a fallos.

Paso 1

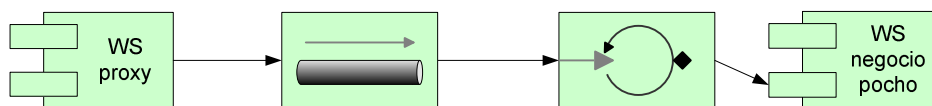
Tendremos un servicio Web SOAP de negocio de un solo sentido que fallará algunas veces, devolviendo una excepción.

Una aplicación cliente de prueba llamará al servicio un número determinado de veces. Sólo algunas de las llamadas tendrán éxito, a causa de los fallos del servicio.

Paso 2

Se desplegará el servicio de negocio en el BUS y se publicará un servicio proxy.

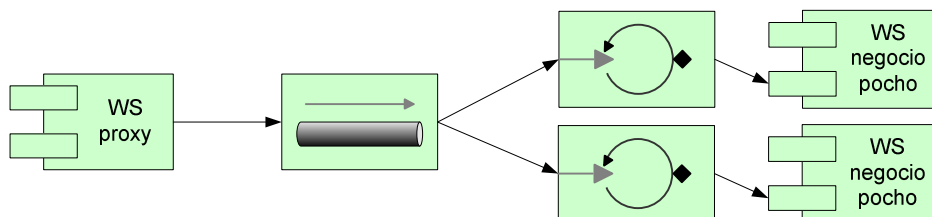
Entre el proxy y el servicio de negocio se implementará un canal de mensajes punto a punto.



La aplicación cliente llamará al proxy, y, a pesar de los fallos del servicio, todas las llamadas tendrán éxito y el servicio de negocio recibirá todos los mensajes.

Paso 3

Se desplegará otra instancia del servicio de negocio y se configurará el canal punto a punto para que haya varios consumidores.



La aplicación cliente llamará al proxy, y, a pesar de los fallos de los servicios, todas las llamadas tendrán éxito, todos los mensajes habrán llegado a una u otra instancia del servicio de negocio y ningún mensaje se habrá recibido dos veces.

CASO DE USO 3. CANAL PUBLICACIÓN/SUBSCRIPCIÓN

Alcance: El objetivo es demostrar la implementación de la integración basada en mensajes con un canal publicador/subscriptor y su tolerancia a fallos.

Paso 1

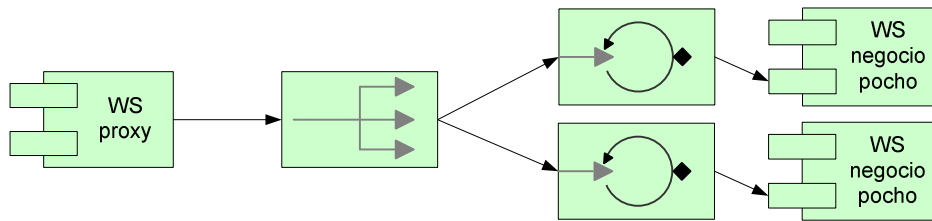
Tendremos dos servicios Web SOAP de negocio de un solo sentido que fallarán algunas veces, devolviendo una excepción. Pueden ser dos instancias del mismo servicio.

Una aplicación cliente de prueba llamará a los dos servicios un número determinado de veces. Sólo algunas de las llamadas tendrán éxito, a causa de los fallos de los servicios.

Paso 2

Se desplegarán los servicios de negocio en el BUS y se publicará un servicio proxy.

Entre el proxy y los servicios de negocio se implementará un canal de mensajes publicador/subscriptor.



La aplicación cliente llamará al proxy para enviar una sola vez cada mensaje y, a pesar de los fallos de los servicios, todas las llamadas tendrán éxito y cada uno de los servicios de negocio recibirá una copia de cada uno de los mensajes.

CASO DE USO 4. ENCAMINAMIENTO BASADO EN EL CONTENIDO

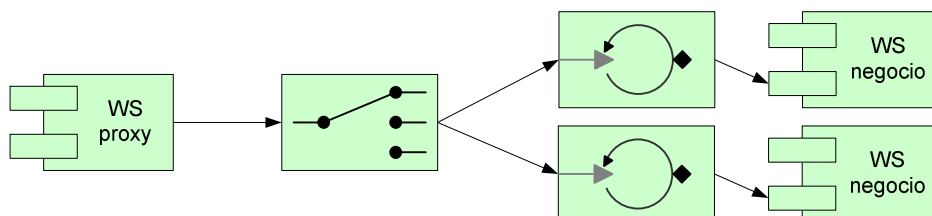
Alcance: El objetivo es demostrar la implementación de un encaminador basado en los contenidos y el impacto de añadir nuevas reglas de encaminamiento.

Paso 1

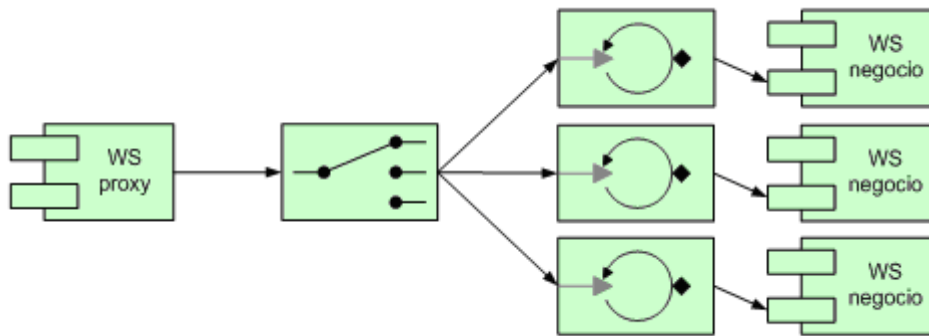
Desplegaremos dos servicios SOAP de negocio de un solo sentido en el bus. Pueden ser dos instancias del mismo servicio.

Una aplicación cliente llamará a un solo servicio proxy, con mensajes de diferente contenido.

Entre el proxy y los servicios de negocio se implementará un encaminador que, dependiendo del contenido de los mensajes, los redirigirá a un servicio de negocio u otro.

**Paso 2**

Se añade un tercer servicio de negocio y se configura el enrutador. La aplicación cliente llamará ahora al proxy con mensajes con tres tipos de contenido diferentes.



CASO DE USO 4. SEPARADOR

Alcance: Tenemos un servicio de negocio SOAP de un solo sentido que acepta mensajes de un tipo. La aplicación cliente envía un agregado de mensajes de ese tipo. El proxy acepta el agregado. Desplegamos el servicio en el bus y entre el proxy y el servicio de negocio se implementa un separador que envía al servicio de negocio los mensajes individuales.

CASO DE USO 5. ESCUCHA EN LLAMADA SÍNCRONA

Alcance: El objetivo es demostrar la implementación de un empalme en una llamada síncrona a un servicio Web desplegado en el bus. La operación debe ser totalmente transparente para la aplicación cliente y el servicio de negocio.

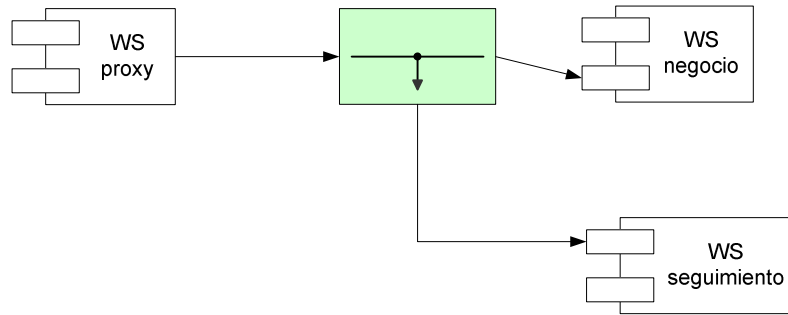
Paso 1

Desplegamos en el bus un servicio de negocio y el correspondiente servicio proxy de conexión directa. Una aplicación cliente llama al proxy normalmente. Ejemplo:



Paso 2

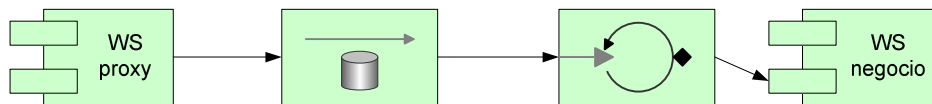
Entre el proxy y el servicio de negocio se despliega un empalme que envía a un segundo servicio de negocio los mensajes de petición y respuesta de las llamadas al servicio de negocio. La aplicación cliente llama al proxy normalmente, y todas las llamadas y respuestas se reenvían al segundo servicio de negocio. Ejemplo:



CASO DE USO 6. CANAL DE ENTREGA GARANTIZADA

Alcance: El objetivo es demostrar la implementación de un canal de mensajes de entrega garantizada, que asegura la entrega de los mensajes aunque el sistema de mensajería falle.

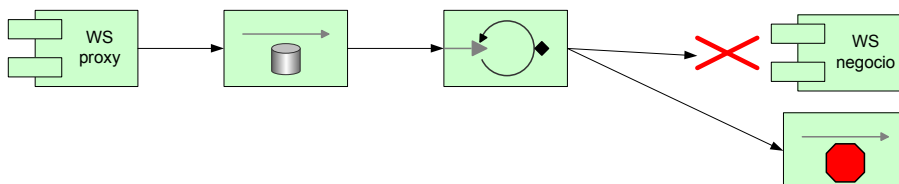
Entre el servicio proxy y el de negocio tendremos un canal de mensajes de entrega garantizada. El servicio de negocio será especialmente lento en el tratamiento de los mensajes.



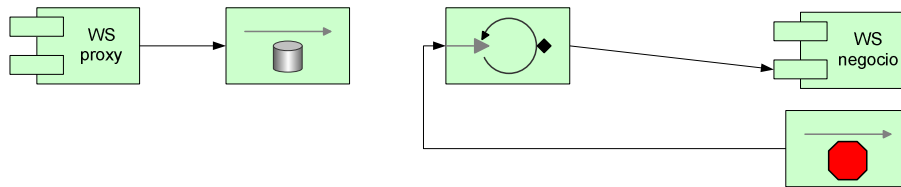
La aplicación cliente llamará al proxy entregando un determinado número de mensajes. Como el consumo de mensajes por parte del servicio de negocio es lento, nos dará tiempo a parar el bus o el servidor de aplicaciones. Después de reiniciarlo, se consumen los mensajes pendientes. Al cabo de un tiempo todos los mensajes han llegado al servicio de negocio.

CASO DE USO 7. CANAL DE MENSAJES FALLIDOS (DEAD LETTER CHANNEL)

Alcance: El caso de uso es una ampliación del canal de entrega garantizada. En primer lugar, cuando el consumidor de mensajes considere que un mensaje no se puede entregar al servicio de negocio (porque falla la conexión con el servicio o porque el propio servicio devuelve una excepción) el mensaje deberá guardarse en un canal de mensajes fallidos.



En segundo lugar, una vez que soporte y operación hayan resuelto el problema que hace que falle la entrega, se podrán reprocesar los mensajes del canal de mensajes fallidos.



Se da por supuesto que el reproceso es una tarea de soporte y operación, es decir, incluirá tareas manuales de reconfiguración y ejecución del reproceso.

Casos de Prueba del Grupo de Arquitectura Desarrollo

CASO DE USO 1. INVOCACIÓN DE WS CON TIPOS DE DATOS PROPIETARIOS DE .NET DESDE JAVA

Existen ciertos WS ya desarrollados en GN en tecnología .net que devuelven tipos de datos no consumibles directamente desde Java y que implican un redesarrollo de dichos WS, no siendo siempre factible este redesarrollo.

El caso de prueba debe seguir y cubrir lo siguiente:

- Crear un WS con las mismas especificaciones que el WS real (GDP) aunque siempre devuelva el mismo dato.
- Publicación del mismo en el Bus e intento de consumirlo y probar a cambiar el tipo si es factible (utilizar todos los recursos de soporte disponibles).
- En caso de no ser posible proponer una alternativa al mismo WS con datos compatibles que devuelva el mismo dato, pero de un modo alternativo y diferente.

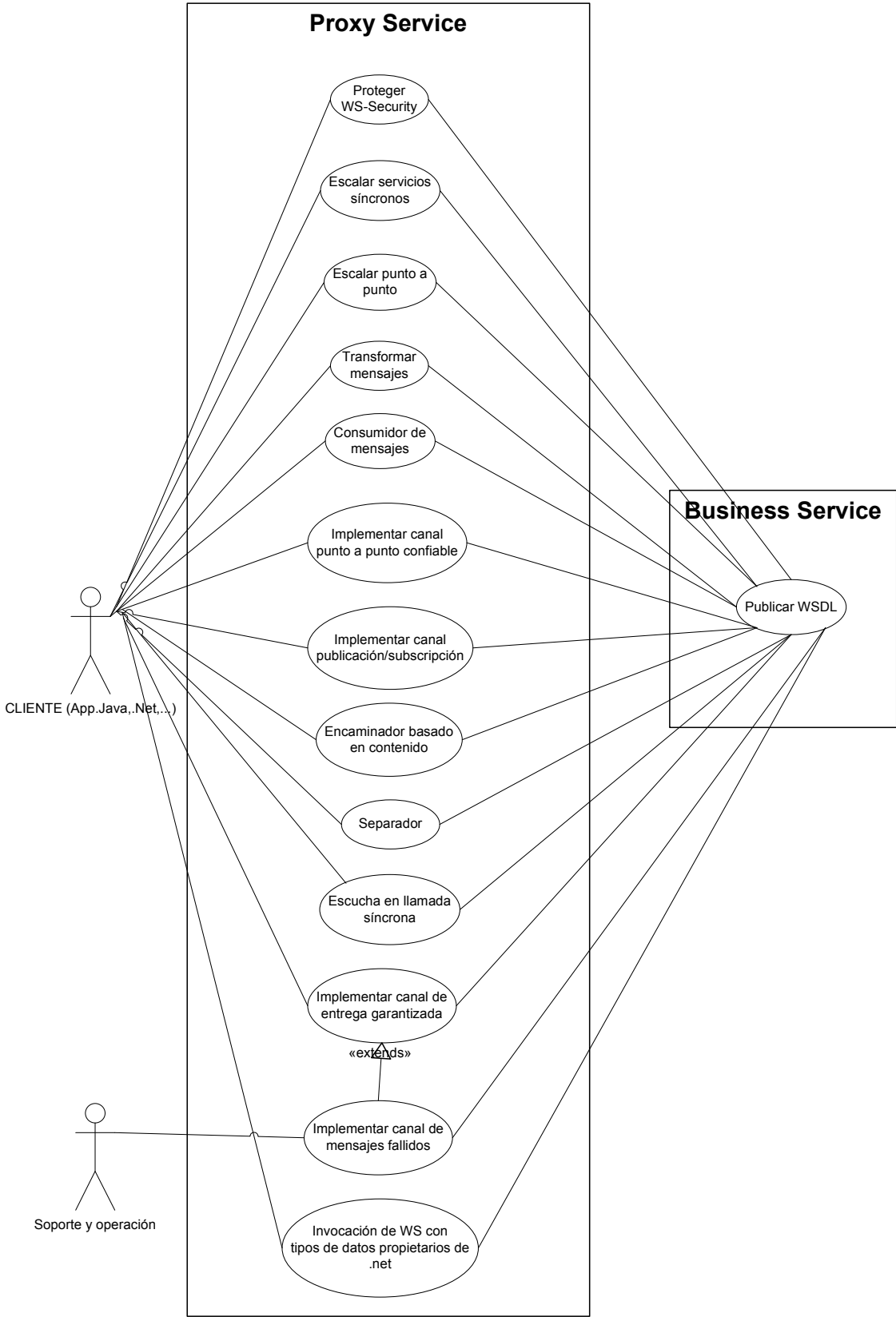
En última instancia, y sólo si está disponible el WS GDP sin datos “no estándar” (en desarrollo actualmente), se intentará publicar esa nueva versión “estándar” en el Bus a modo de validación funcional de que el Bus permitirá su publicación.

Modelo y Análisis de casos de uso

En este apartado se va a realizar una pequeña descripción de cada caso de uso, con su diagrama de caso de uso correspondiente y con su diagrama de actividad en los casos que se requiera.

Se van a desarrollar los dos entregables de diseño y 13 casos de uso de 15 existentes. La razón por la que no se desarrollan todos los casos de uso es porque en un principio existían 13 casos pero los requisitos se han ido cambiando conforme avanza el proyecto, es decir, se han añadido dos nuevos casos. Por esta razón, los solicitantes han preferido desarrollar estos dos nuevos casos y dejar los siguientes casos sin implementar ya que no los consideran importantes: Caso de uso 2 (GAF) publicar y proteger con autenticación integrada un WS sin proteger y caso de uso 6 (GAF) despliegue de un proceso de negocio CES en el bus

En el diagrama de casos de uso de la Figura 6 se representan todas las funcionalidades a implementar en la aplicación.

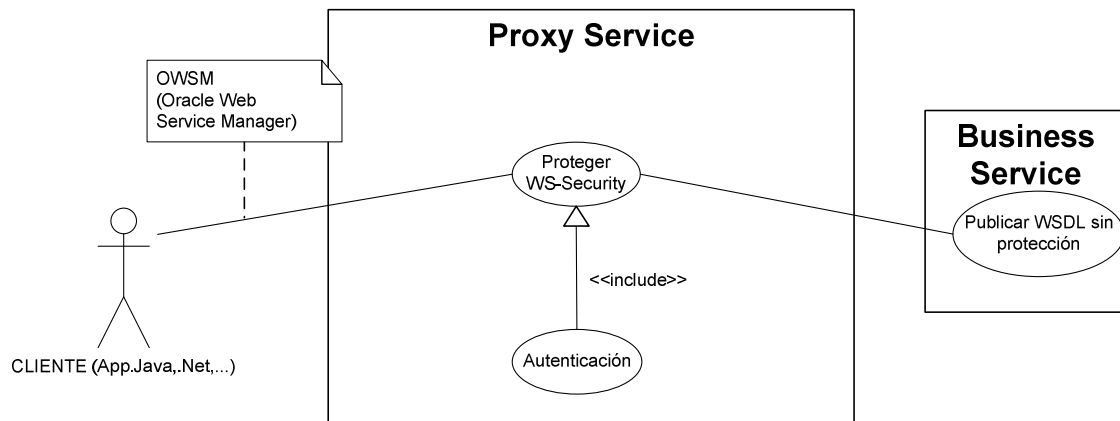


-Figura 6.Diagrama de Casos de Uso general-

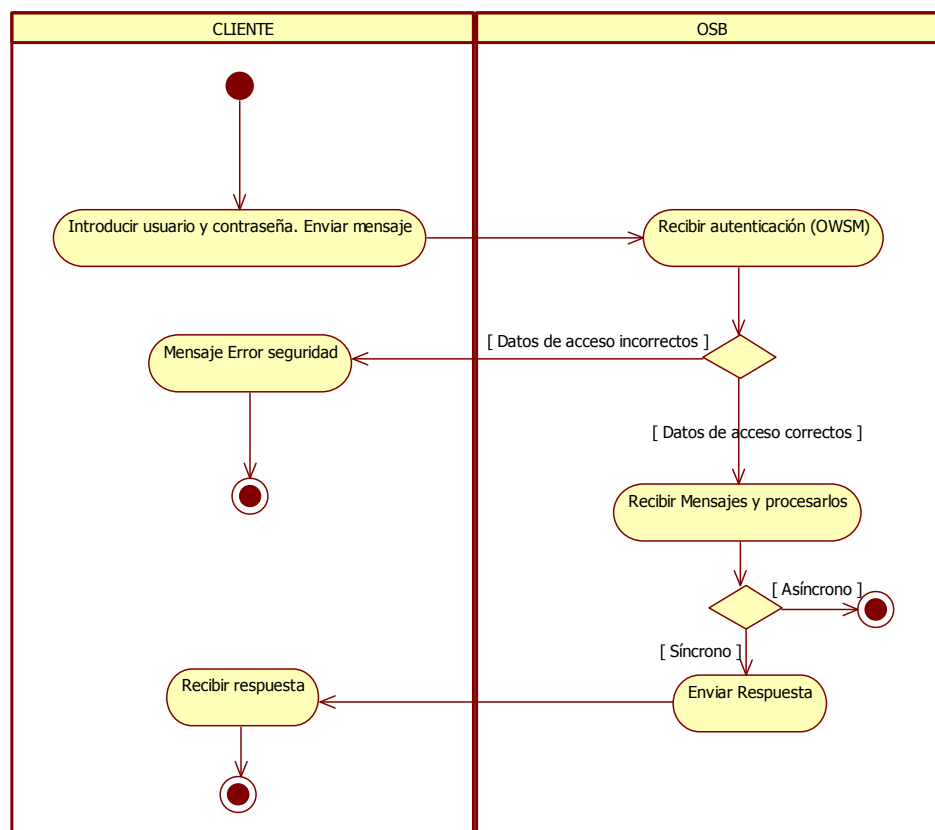
A continuación se hace una descripción de cada uno de los casos de uso del diagrama anterior.

Casos de Prueba del Grupo de Arquitectura Funcional

1. Caso de uso publicar y proteger con *WS-Security* un WS sin protección: Una aplicación cliente desea acceder a un *Web Service* sin protección, protegido por *WS-Security* por el OSB. Una vez realizada la verificación del usuario, se envía el mensaje al servicio de negocio y la aplicación cliente recibe la respuesta.



-Figura 7. Diagrama de casos de uso. GAF Caso 1-

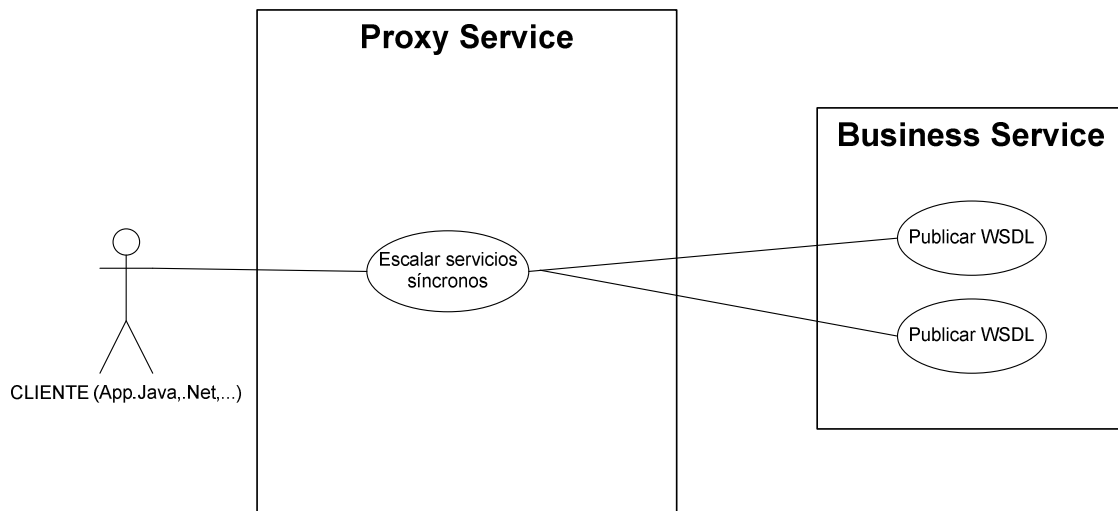


-Figura 8. Diagrama de actividad. GAF Caso 1-

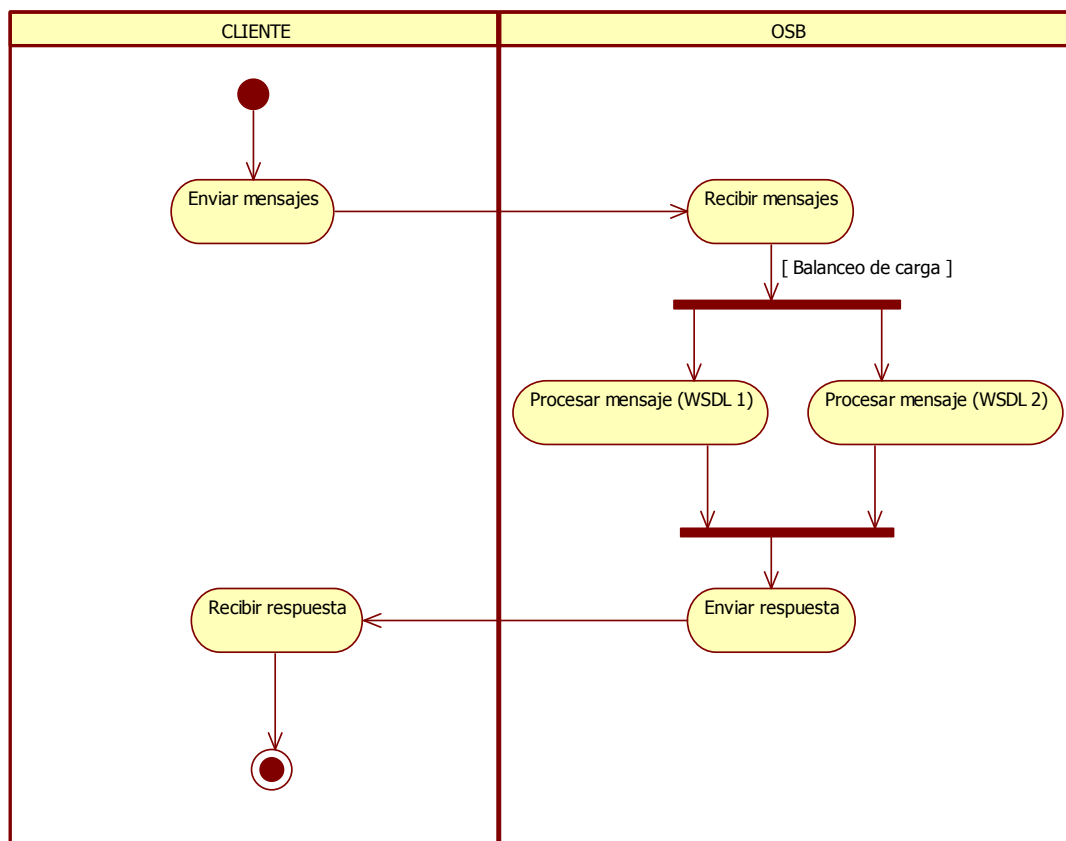
CASO DE USO 1. PUBLICAR Y PROTEGER CON WS-SECURITY UN WS SIN PROTECCIÓN	
Responsable-Persona de contacto	GAF (Grupo de Arquitectura Funcional)
Breve descripción (más información en el apartado análisis de requisitos)	Hacer que el OSB publique un <i>Proxy Service</i> protegido con <i>WS-Security (UserNameToken)</i> . El <i>Business Service</i> sería un <i>Web Service</i> abierto, sin protección.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ul style="list-style-type: none"> • Código fuente del proyecto de prueba en repositorio de fuentes. • Solución desplegada en el bus de laboratorio. • Desplegable e Instrucciones de despliegue. • Informe de resultados de las pruebas. <ul style="list-style-type: none"> ○ Probar que el OSB llama al <i>Web Service</i> abierto. ○ Probar que el OSB publica hacia fuera el nuevo <i>Web Service</i>. ○ Probar que desde .NET podemos invocar el WS del OSB, con la autenticación integrada Windows (<i>Negotiate</i>) y que no podemos sin pasárselas. ○ Probar que desde la aplicación Java podemos invocar el WS pasándole las credenciales y que no podemos sin pasárselas. • Documento de detalle de la implementación.

-Tabla 4. Descripción GAF Caso 1-

2. Caso de uso escalabilidad en servicios síncronos: una aplicación cliente envía varios mensajes al OSB. El *Proxy Service* es el encargado de dividir la carga entre los diferentes servicios de negocio.



-Figura 9-Diagrama de casos de uso. GAF Caso 2-

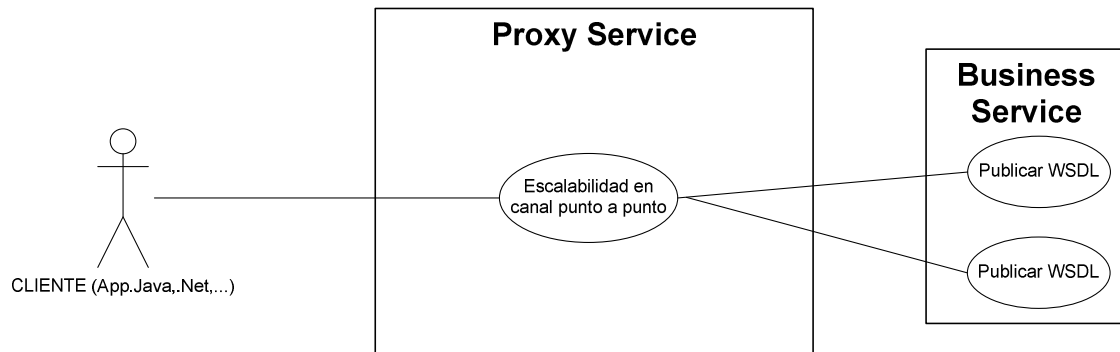


-Figura 10.Diagrama de actividad. GAF Caso 2-

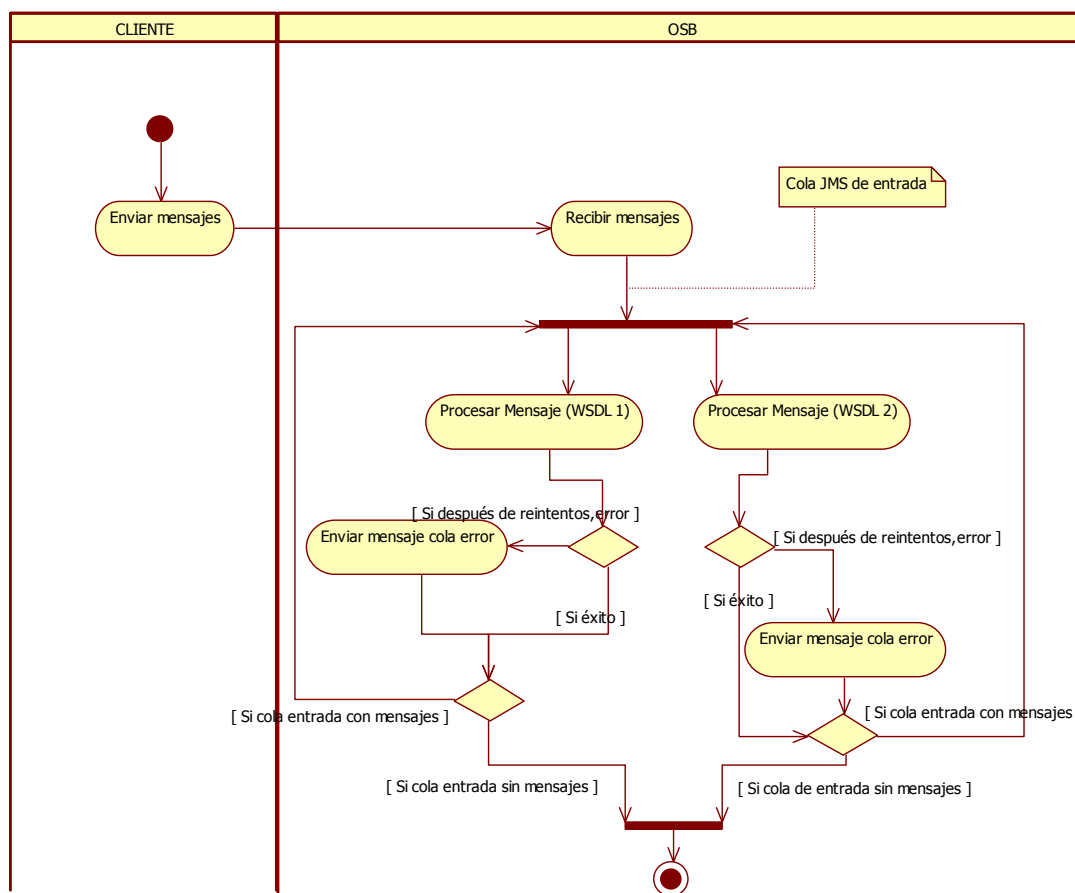
CASO DE USO 2. ESCALABILIDAD EN SERVICIOS SINCRONOS	
Responsable-Persona de contacto	GAF (Grupo de Arquitectura Funcional)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la escalabilidad que proporciona contar con varios servicios de negocio iguales.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. <ol style="list-style-type: none"> i. Paso 1. ii. Paso 2. 5. Documento de detalle de la implementación.

-Tabla 5. Descripción GAF Caso 2-

3. Caso de uso escalabilidad en canal punto a punto: una aplicación cliente envía varios mensajes al OSB. El *Proxy Service* es el encargado de dividir la carga entre los diferentes servicios de negocio. A diferencia del caso anterior, el cliente no espera respuesta (asíncrono) y se implementa con colas JMS.



-Figura 11. Diagrama de casos de uso. GAF Caso 3-

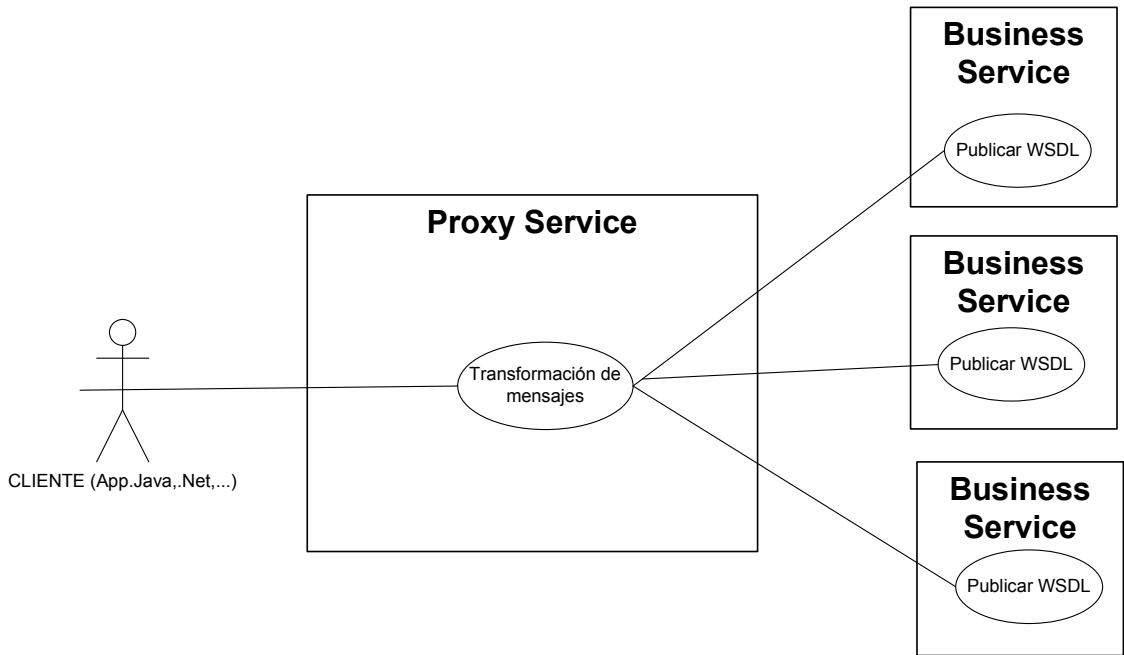


-Figura 12. Diagrama de actividad. GAF Caso 3-

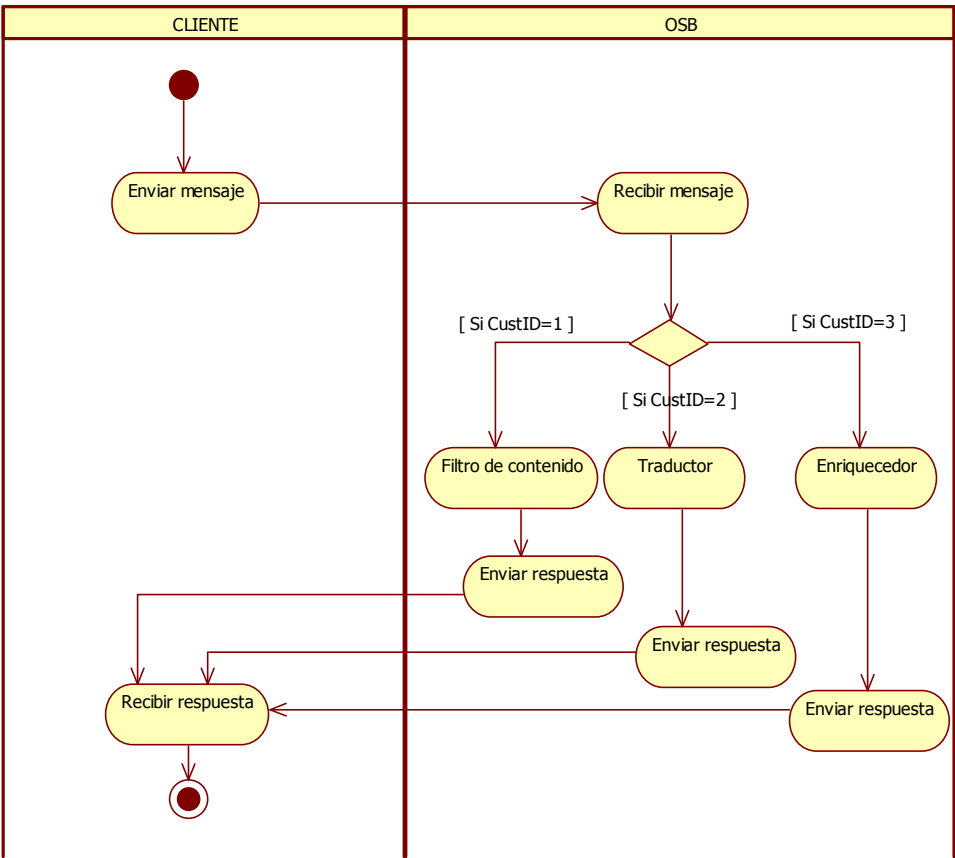
CASO DE USO 3. ESCALABILIDAD EN CANAL PUNTO A PUNTO	
Responsable-Persona de contacto	GAF (Grupo de Arquitectura Funcional)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la escalabilidad que proporciona un canal punto a punto con varios consumidores iguales.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. <ol style="list-style-type: none"> i. Paso 1. ii. Paso 2. 5. Documento de detalle de la implementación.

-Tabla 6. Descripción GAF Caso 3-

4. Caso de uso transformación de mensajes: permite transformar los mensajes de entrada según las necesidades del servicio de negocio.



-Figura 13. Diagrama de casos de uso. GAF Caso 4-



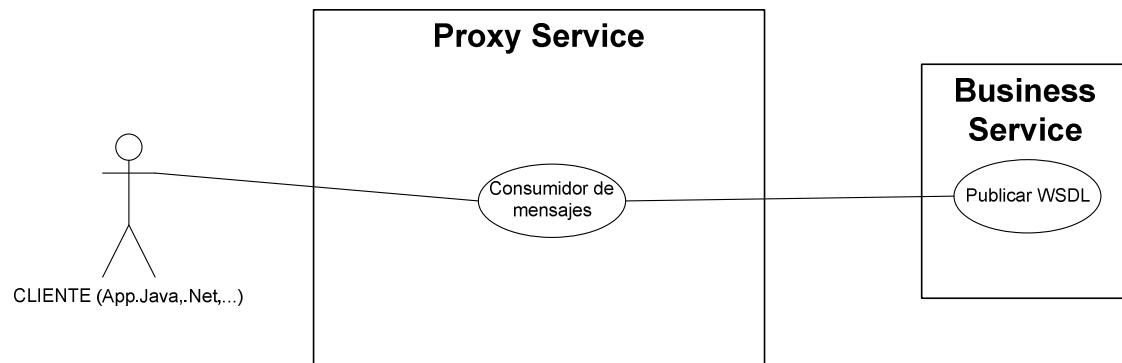
-Figura 14. Diagrama de actividad. GAF Caso 4-

CASO DE USO 4. TRANSFORMACIÓN DE MENSAJES	
Responsable-Persona de contacto	GAF (Grupo de Arquitectura Funcional)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la transformación de mensajes que proporciona el bus.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. <ol style="list-style-type: none"> i. Paso 1. ii. Paso 2. i. Paso3. 5. Documento de detalle de la implementación.

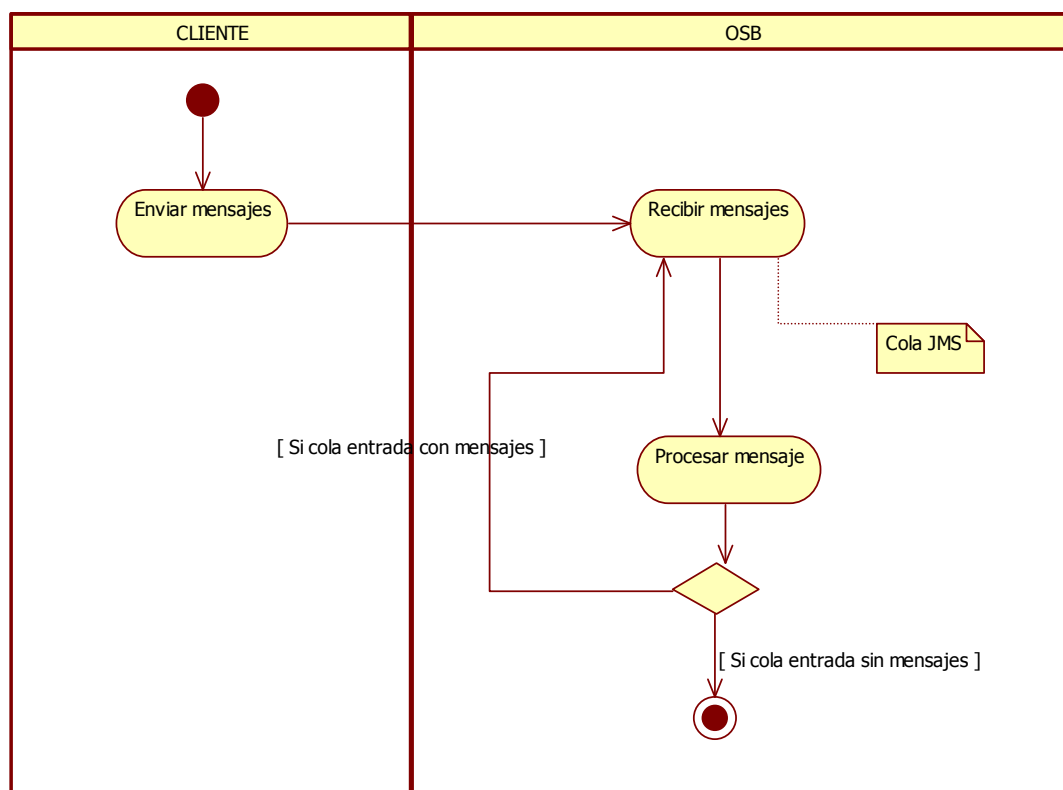
-Tabla 7. Descripción GAF Caso 4-

Casos de Prueba de la Sección de Procedimientos Electrónicos

1. Caso de uso consumidor de mensajes (*Polling consumer*): permite a la aplicación cliente enviar mensajes sin esperar respuesta (asíncrono) y que los mensajes se vayan almacenando en una cola JMS para poder ser tratados cuando el servicio de negocio se encuentre disponible.



-Figura 15. Diagrama de casos de uso. SPE Caso 1-

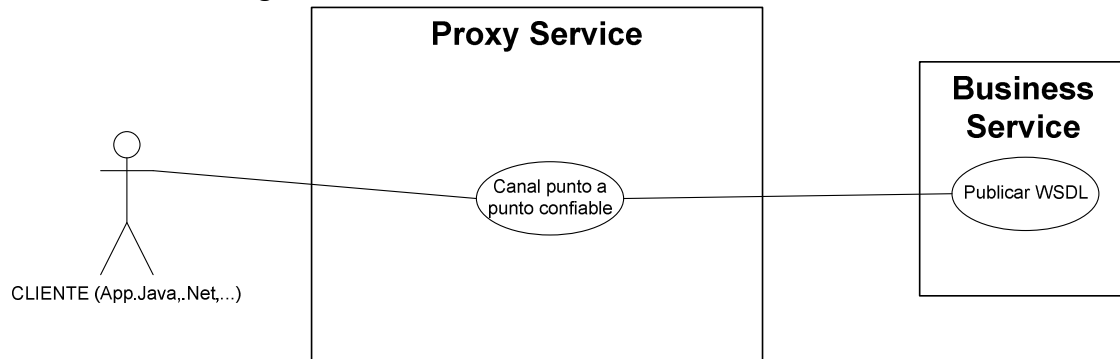


-Figura 16. Diagrama de actividad. SPE Caso 1-

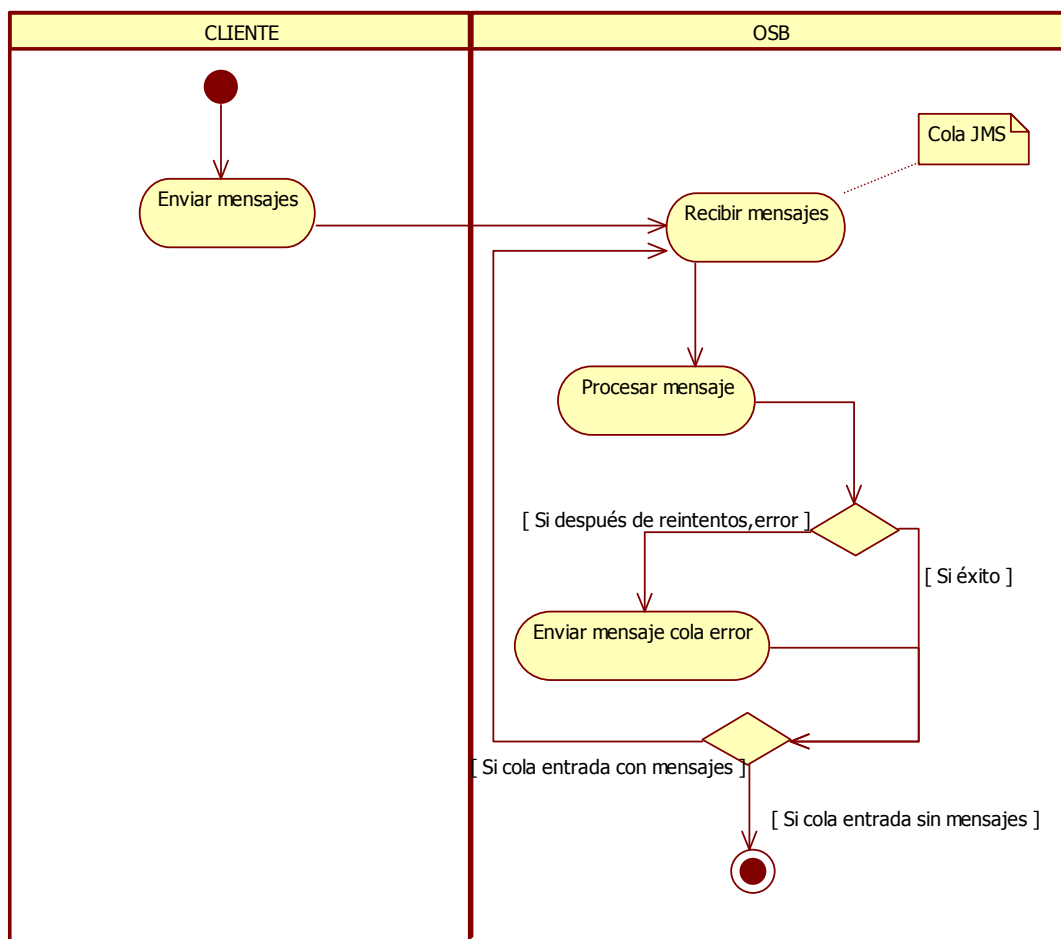
CASO DE USO 1. CONSUMIDOR DE MENSAJES (POLLING CONSUMER)	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de un consumidor de mensajes que actúe de intermediario entre un servicio Web de un solo sentido y una cola de mensajes.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

-Tabla 8. Descripción SPE Caso 1-

2. Caso de uso canal punto a punto confiable: permite a una aplicación cliente el envío de mensajes sin recibir respuesta (asíncrono) de manera que si el servicio de negocio al que llamamos falla, los mensajes no se pierdan, es decir, se guarden en una cola JMS de error.



-Figura 17 . Diagrama de casos de uso. SPE Caso 2-

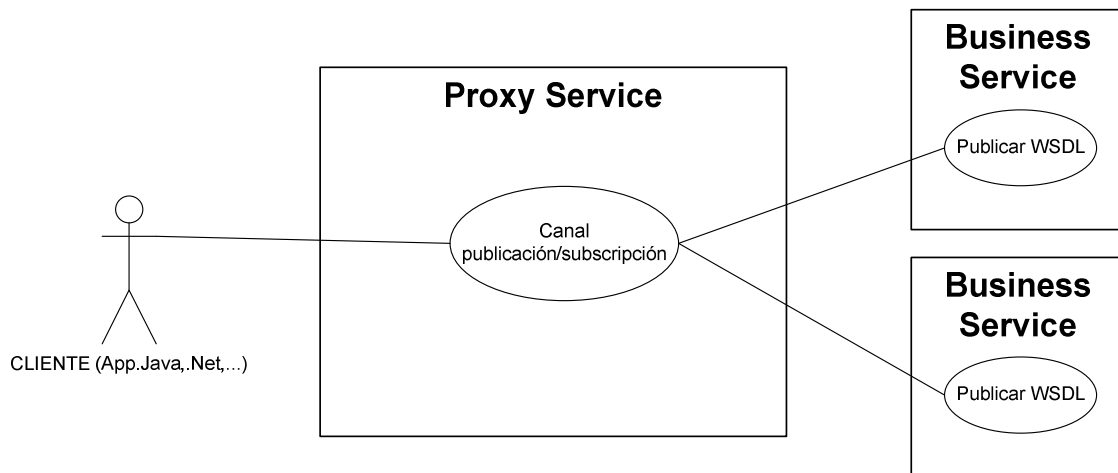


-Figura 18. Diagrama de actividad. SPE Caso 2-

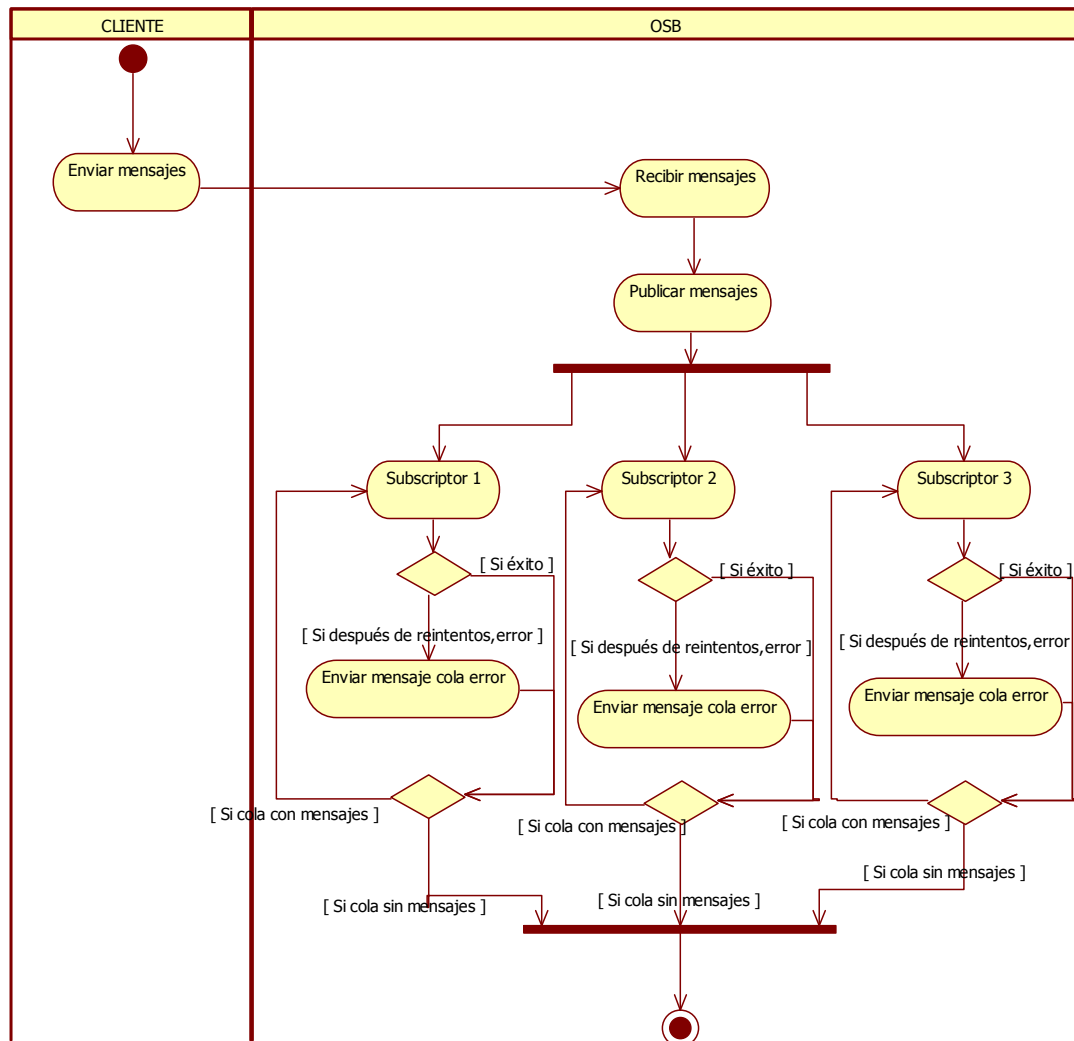
CASO DE USO 2. CANAL PUNTO A PUNTO CONFIABLE	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de la integración basada en mensajes con un canal punto a punto y su tolerancia a fallos.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. <ol style="list-style-type: none"> i. Paso 1. ii. Paso 2. iii. Paso 3. 5. Documento de detalle de la implementación.

-Tabla 9. Descripción SPE Caso 2-

3. Caso de uso canal de publicación / subscripción: permite enviar un mismo mensaje a varios subscriptores.



-Figura 19. Diagrama de casos de uso. SPE Caso 3-

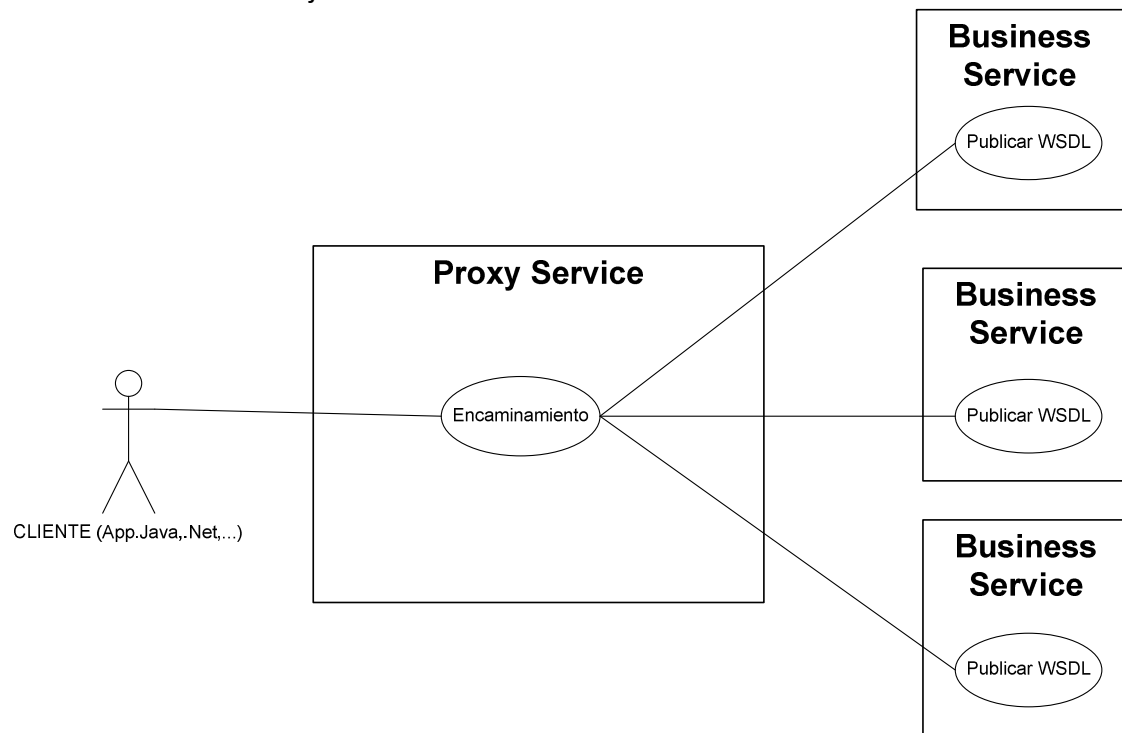


-Figura 20. Diagrama de actividad. SPE Caso 3-

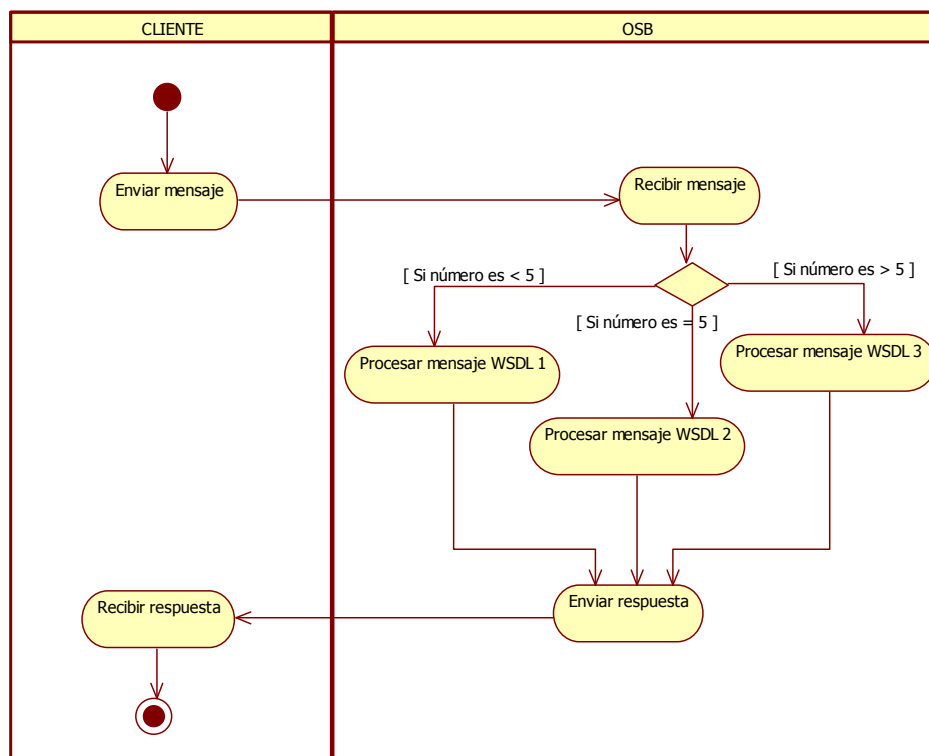
CASO DE USO 3. CANAL PUBLICACIÓN/SUBSCRIPCIÓN	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de la integración basada en mensajes con un canal publicador/subscriptor y su tolerancia a fallos.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 1. Solución desplegada en el bus de laboratorio. 2. Desplegable e Instrucciones de despliegue. 3. Informe de resultados de las pruebas. <ol style="list-style-type: none"> i. Paso 1. ii. Paso 2. 4. Documento de detalle de la implementación.

-Tabla 10. Descripción SPE Caso 3-

4. Caso de uso encaminamiento basado en el contenido: Permite encaminar un mensaje a diferentes servicios de negocio dependiendo del contenido de dicho mensaje.



-Figura 21. Diagrama de casos de uso. SPE Caso 4-



-Figura 22. Diagrama de actividad. SPE Caso 4-

CASO DE USO 4. ENCAMINAMIENTO BASADO EN EL CONTENIDO	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de un encaminador basado en los contenidos y el impacto de añadir nuevas reglas de encaminamiento.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. <ol style="list-style-type: none"> a. Paso 1. b. Paso 2, haciendo incapié en el impacto de añadir nuevas reglas de encaminamiento. 5. Documento de detalle de la implementación.

-Tabla 11. Descripción SPE Caso 4-

5. Caso de uso separador: Permite a la aplicación cliente enviar varios ítems juntos y es el *Proxy Service* el encargado de dividir cada uno de los ítems y procesarlos por separado.

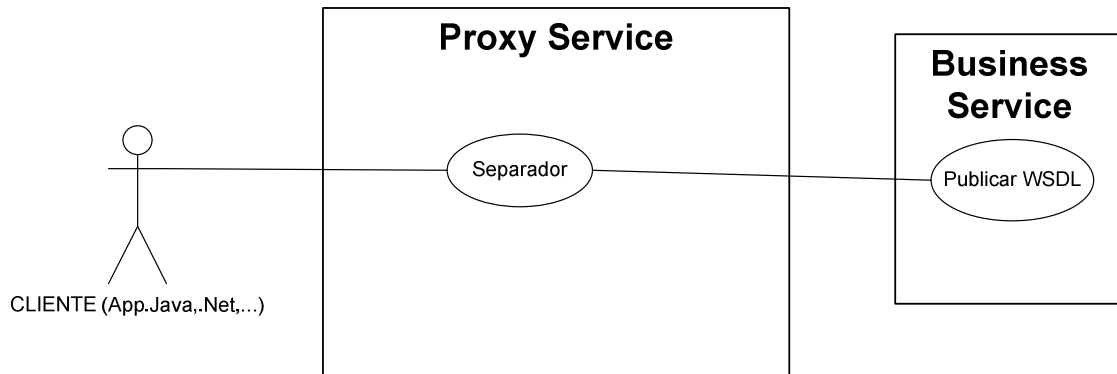
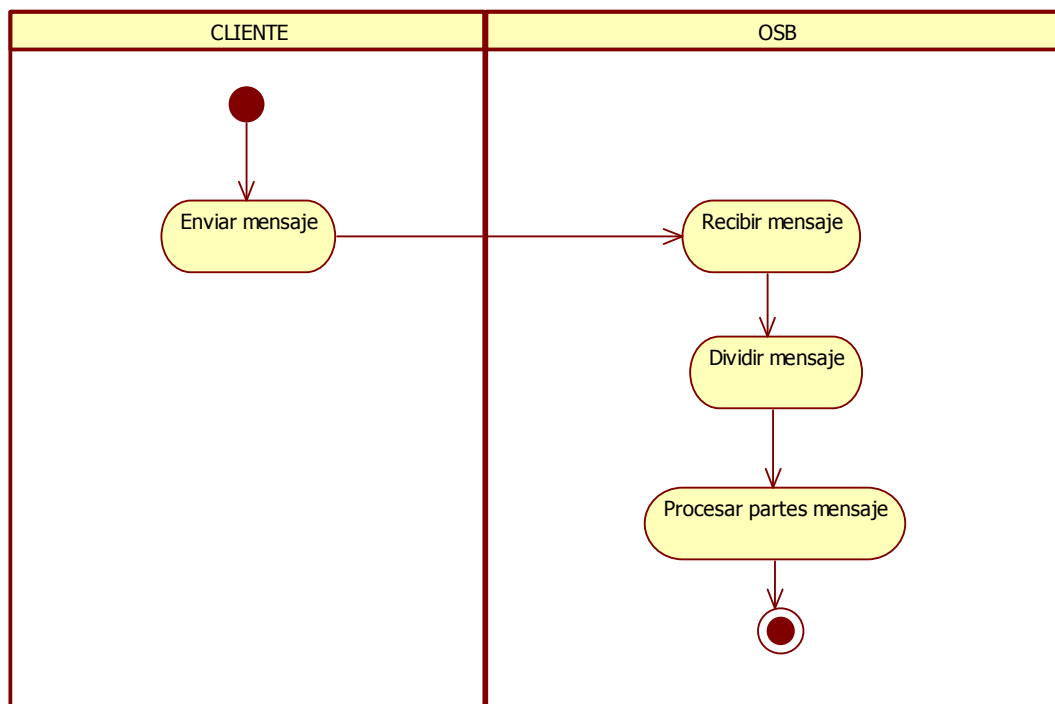


Figura 23. Diagrama de casos de uso. SPE Caso 5

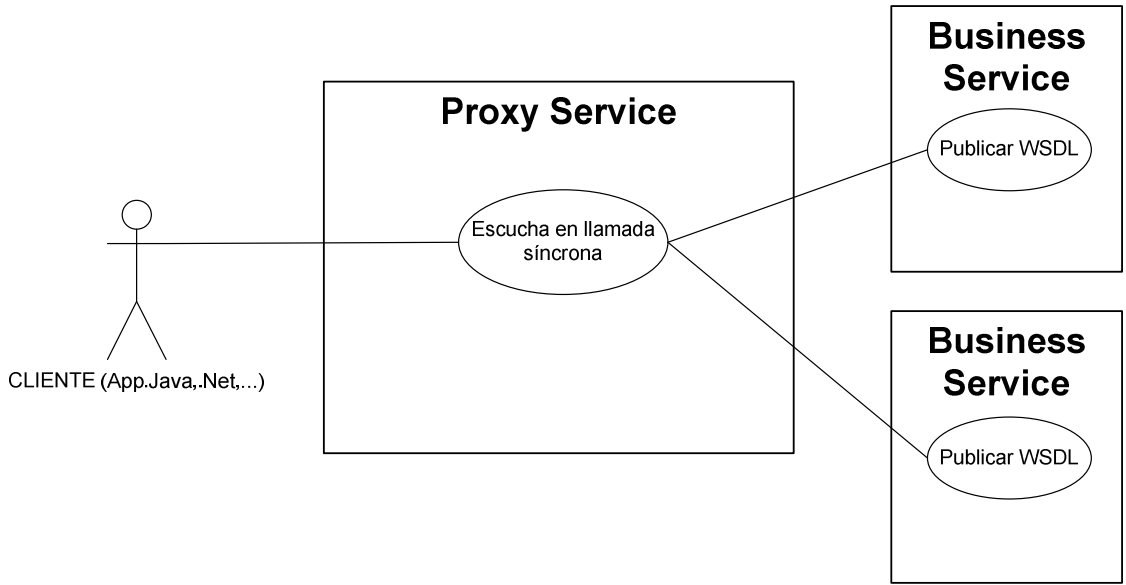


-Figura 24. Diagrama de actividad. SPE Caso 5-

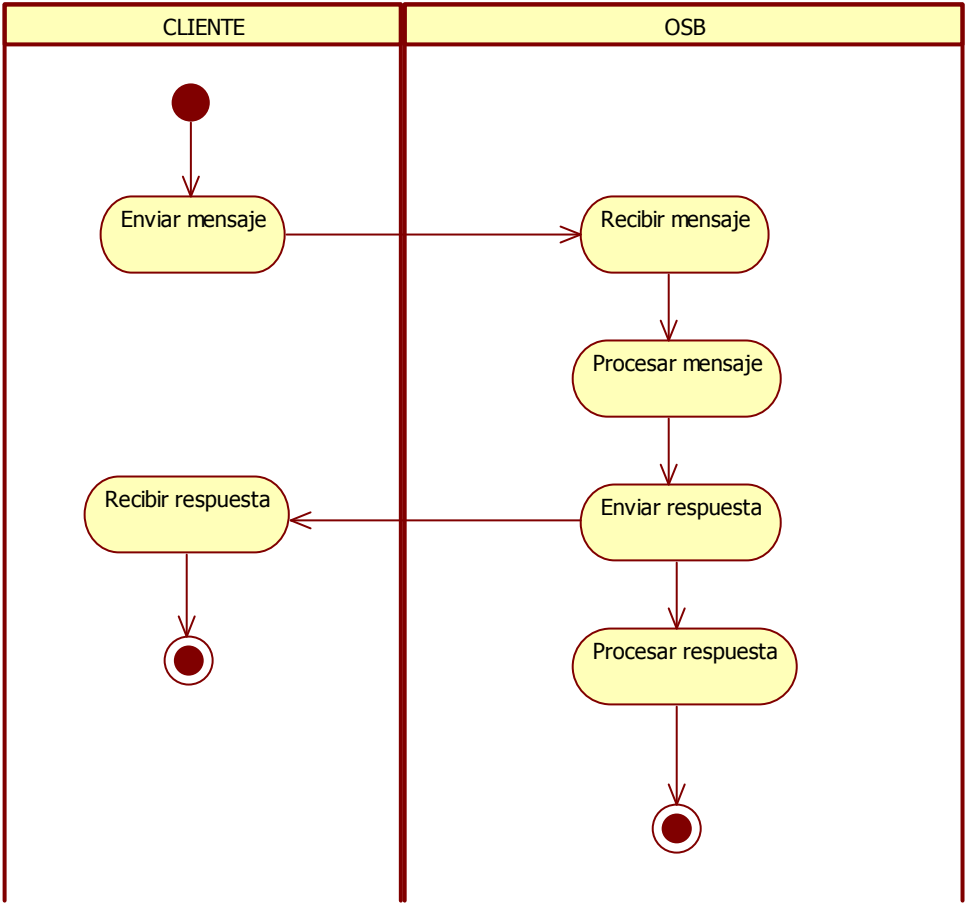
CASO DE USO 5. SEPARADOR	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	Tenemos un servicio de negocio SOAP de un solo sentido que acepta mensajes de un tipo. La aplicación cliente envía un agregado de mensajes de ese tipo. El proxy acepta el agregado. Desplegamos el servicio en el bus y entre el proxy y el servicio de negocio se implementa un separador que envía al servicio de negocio los mensajes individuales.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

-Tabla 12. Descripción SPE Caso 5-

6. Caso de uso escucha en llamada síncrona: Permite a una aplicación cliente enviar un mensaje al servicio de negocio y obtener una respuesta. De manera transparente, tanto el mensaje de petición como la respuesta son enviados a un segundo servicio de negocio.



-Figura 25. Diagrama de casos de uso. SPE Caso 6-

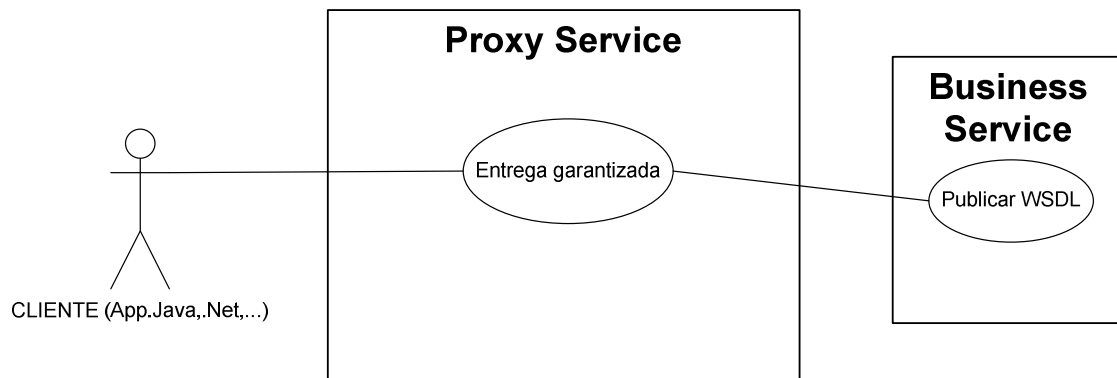


-Figura 26. Diagrama de actividad. SPE Caso 6-

CASO DE USO 6. ESCUCHA EN LLAMADA SÍNCRONA	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de un empalme en una llamada síncrona a un servicio Web desplegado en el bus. La operación debe ser totalmente transparente para la aplicación cliente y el servicio de negocio.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

-Tabla 13. Descripción SPE Caso 6-

7. Caso de uso canal de entrega garantizada: Asegura que los mensajes enviados por una aplicación cliente sean entregados a los servicios de negocio correspondiente aunque el sistema falle.

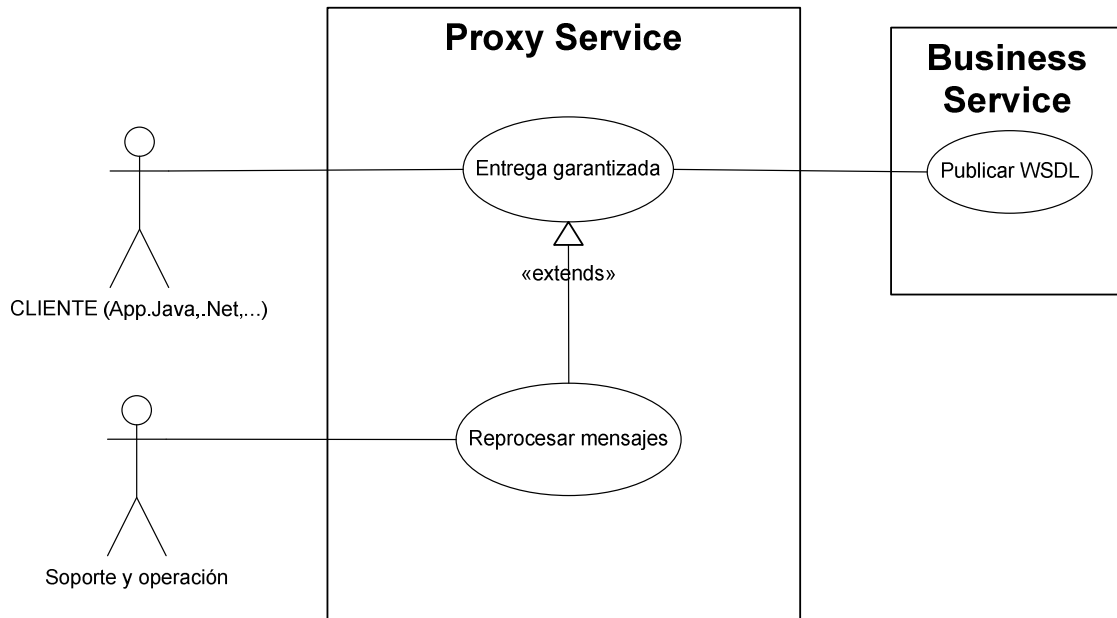


-Figura 27. Diagrama de casos de uso. SPE Caso 7-

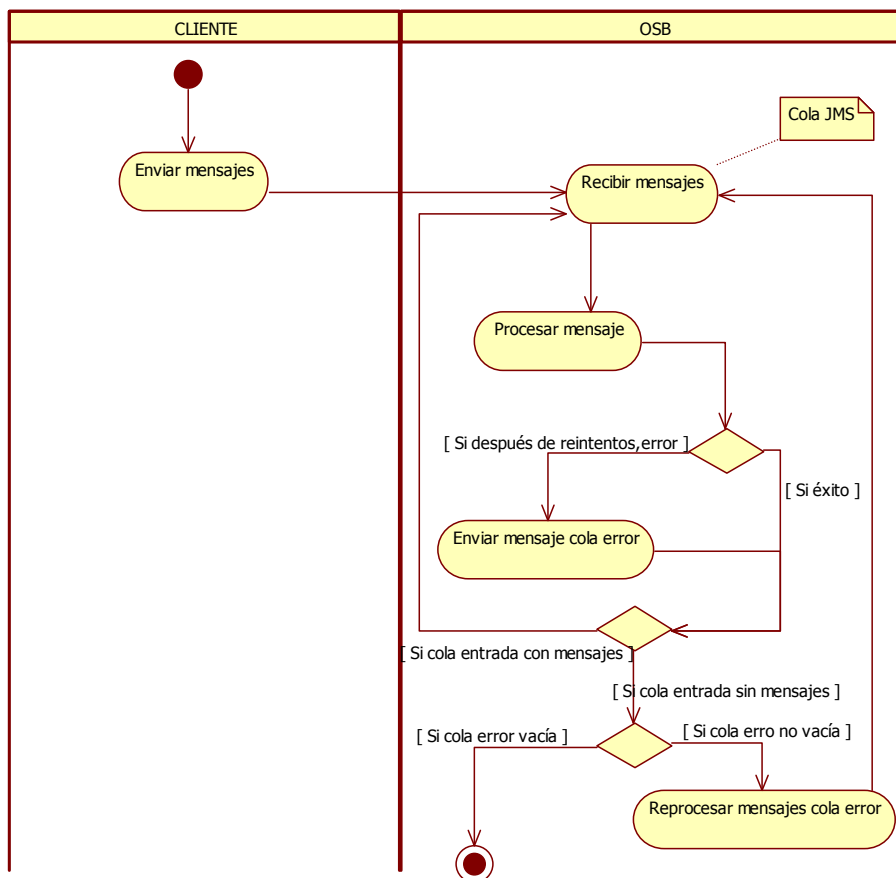
CASO DE USO 7. CANAL DE ENTREGA GARANTIZADA	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es demostrar la implementación de un canal de mensajes de entrega garantizada, que asegura la entrega de los mensajes aunque el sistema de mensajería falle.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

-Tabla 14. Descripción SPE Caso 7-

8. Caso de uso canal de mensajes fallidos (dead letter channel): Permite reprocesar los mensajes que se encuentran en la cola de error una vez que se han solventado los problemas del servicio de negocio.



-Figura 28. Diagrama de casos de uso. SPE Caso 8-



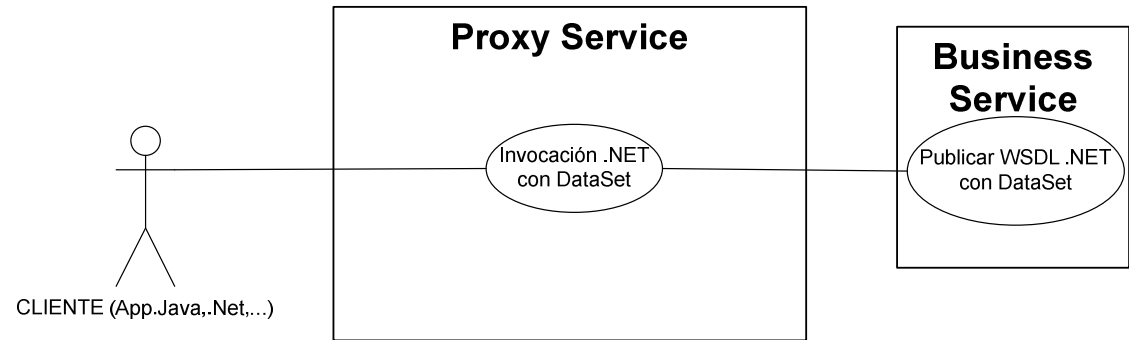
-Figura 29. Diagrama de actividad. SPE Caso 8-

CASO DE USO 8. CANAL DE MENSAJES FALLIDOS (DEAD LETTER CHANNEL)	
Responsable-Persona de contacto	SPE (Sección de Procedimientos Electrónicos)
Breve descripción (más información en el apartado análisis de requisitos)	El caso de uso es una ampliación del canal de entrega garantizada. Una vez que soporte y operación hayan resuelto el problema que hace que falle la entrega, se podrán reprocesar los mensajes del canal de mensajes fallidos.
Elementos o entregables para iniciar el trabajo	Ninguno. Los servicios Web de negocio serán servicios simples desarrollados a medida.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

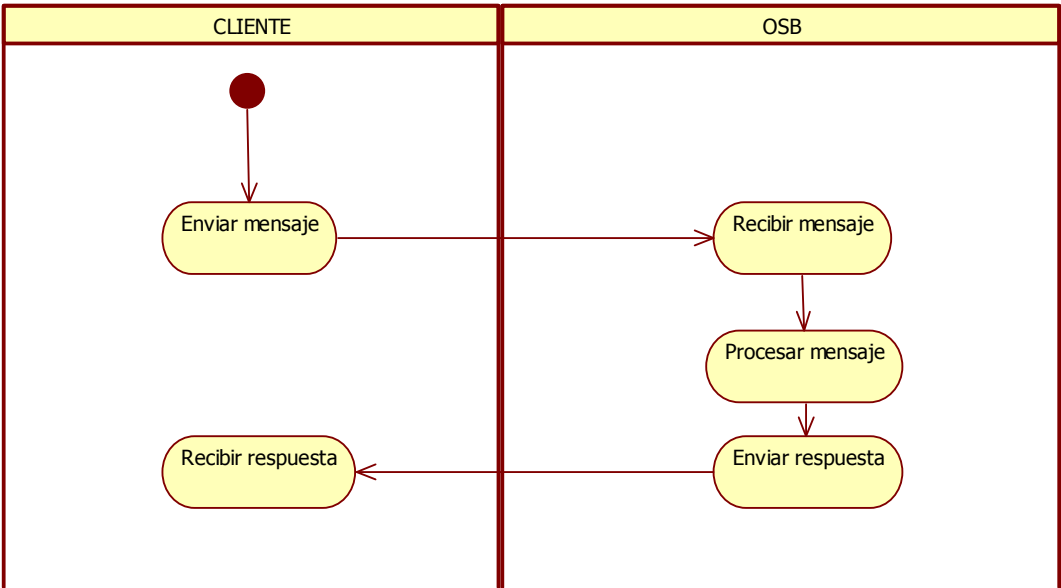
-Tabla 15. Descripción SPE Caso 8-

Casos de Prueba del Grupo de Arquitectura de Desarrollo

- 1. Caso de uso invocación de WS con tipos de datos propietarios de .Net desde Java: Permite a una aplicación cliente Java utilizar un *Web Service* con tipo de datos *DataSet*.



-Figura 30. Diagrama de casos de uso. GAD Caso 1-



-Figura 31. Diagrama de actividad. GAD Caso 1-

CASO DE USO 1. INVOCACIÓN DE WS CON TIPOS DE DATOS PROPIETARIOS DE .NET DESDE JAVA	
Responsable-Persona de contacto	GAD (Grupo de Arquitectura de Desarrollo) La prueba se hará con un WS desarrollado en el proyecto, pero siguiendo las especificaciones de un <i>Web Service</i> real: El Gestor de Direcciones Postales (GDP).
Breve descripción (más información en el apartado análisis de requisitos)	Existen ciertos WS ya desarrollados en GN en tecnología .net que devuelven tipos de datos no consumibles directamente desde Java y que implican un redesarrollo de dichos WS, no siendo siempre factible este redesarrollo.
Elementos o entregables para iniciar el trabajo	Documentación del servicio web de Direcciones Postales. WSDL del Servicio Web “estándar”.
Criterio de finalización	<ol style="list-style-type: none"> 1. Código fuente del proyecto de prueba en repositorio de fuentes. 2. Solución desplegada en el bus de laboratorio. 3. Desplegable e Instrucciones de despliegue. 4. Informe de resultados de las pruebas. 5. Documento de detalle de la implementación.

-Tabla 16. Descripción GAD Caso 1-

Entregables de diseño del Grupo de Arquitectura de Infraestructuras

ENTREGABLE DE DISEÑO 1. INFORME DE CAPACIDADES DE INFORMES Y ALERTAS CON ORACLE SERVICE BUS	
Responsable-Persona de contacto	GAI (Grupo de Arquitectura de Infraestructuras)
Breve descripción (más información en el apartado análisis de requisitos)	Conocer las capacidades de monitorización que da OSB.
Elementos o entregables para iniciar el trabajo	
Criterio de finalización	<p>Documento con el siguiente contenido:</p> <ul style="list-style-type: none"> • Tipos de Informes que se pueden obtener por defecto en la herramienta así como los mecanismos de obtención (ejecución bajo demanda, ejecución programada, etc.). • Capacidades de desarrollo de informes. • Valores de medición: descripción de los valores de medición sobre los servicios publicados en el bus: valores de disponibilidad, valores de rendimiento, valores de capacidad. • Capacidades de monitorización y Alertas: capacidades de la herramienta para enviar alertas (qué valores de medida pueden lanzar alertas), así como mecanismos de envío de alertas: Protocolos (¿snmp?, correo electrónico, ¿otros?).

-Tabla 17. Descripción GAI Entregable 1-

ENTREGABLE DE DISEÑO 2. PROPUESTA DE PROCESO DE GESTIÓN DE CAMBIOS CON ORACLE SERVICE BUS	
Responsable-Persona de contacto	GAI (Grupo de Arquitectura de Infraestructuras)
Breve descripción (más información en el apartado análisis de requisitos)	El objetivo es estudiar los pasos necesarios para la incorporación del Bus en el proceso de gestión de cambio y despliegues actual en Gobierno de Navarra.
Elementos o entregables para iniciar el trabajo	Documentos/información que describen los procesos de Gestión de Cambio en Gobierno de Navarra. Documentos/información que describen el ciclo de gestión de proyectos de Gobierno de Navarra. Documentación/información sobre el entorno de integración continua y ciclo de vida de desarrollo.
Criterio de finalización	Documento que describa: <ul style="list-style-type: none"> • Dónde y cómo se incorpora el bus en el ciclo de despliegue: procedimiento detallado del proceso de despliegue en el bus en los diferentes entornos (desarrollo, integración, preproducción y producción). • Descripción de los entregables de un proyecto una vez integrado el bus de servicios: tipo de empaquetado de despliegue. • Necesidades y características de los diferentes entornos.

-Tabla 18. Descripción GAI Entregable 2-

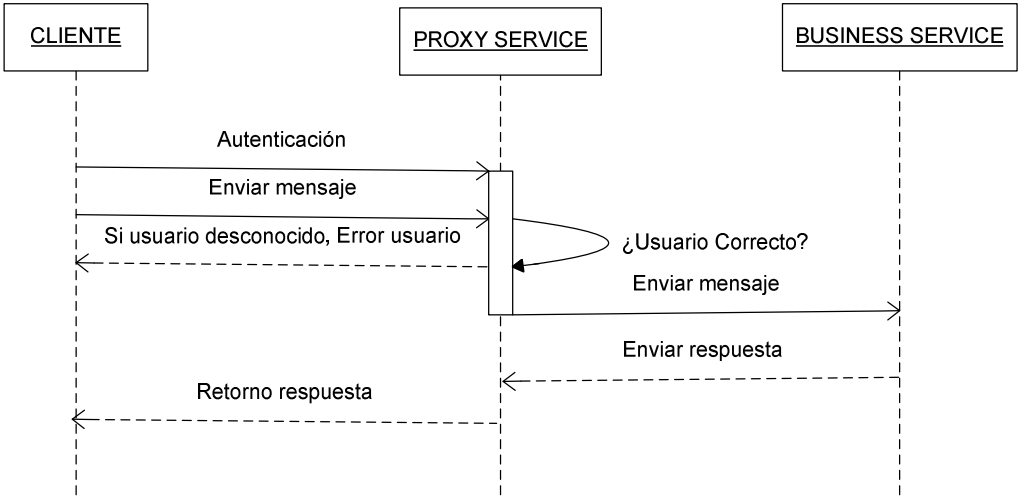
Diseño

Diagramas de secuencia

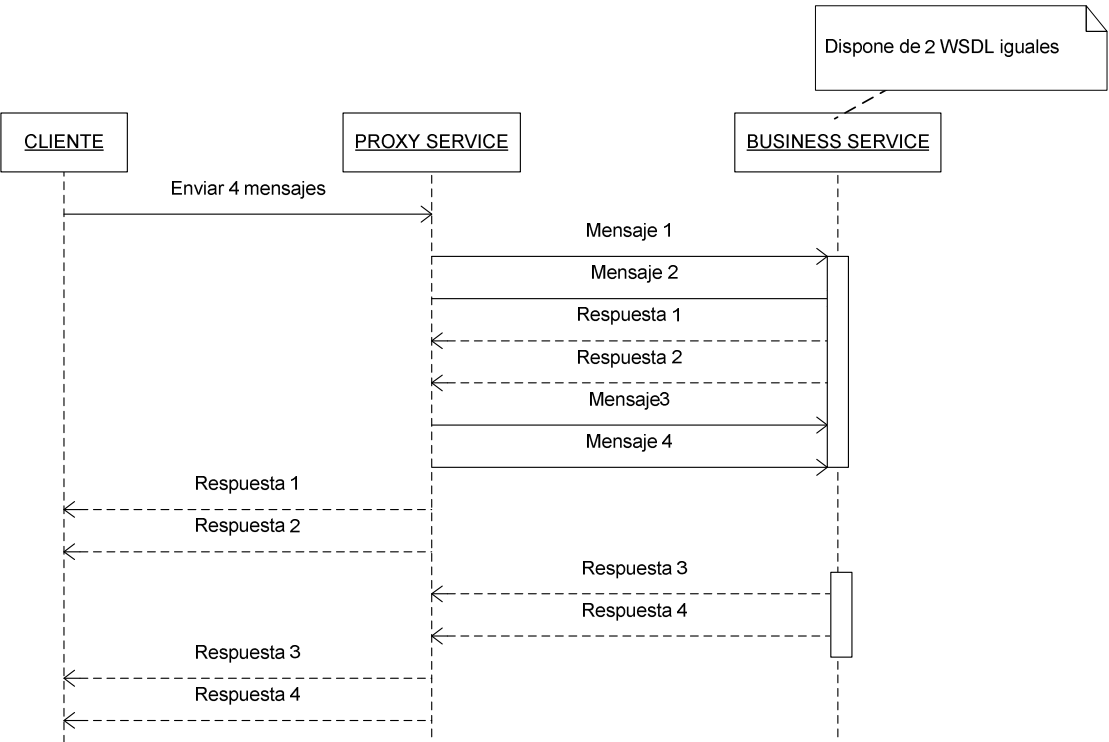
En este apartado se representan los diagramas de secuencia que representan cada uno de los casos de uso. Estos diagramas permiten mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones.

Casos de Prueba del Grupo de Arquitectura Funcional

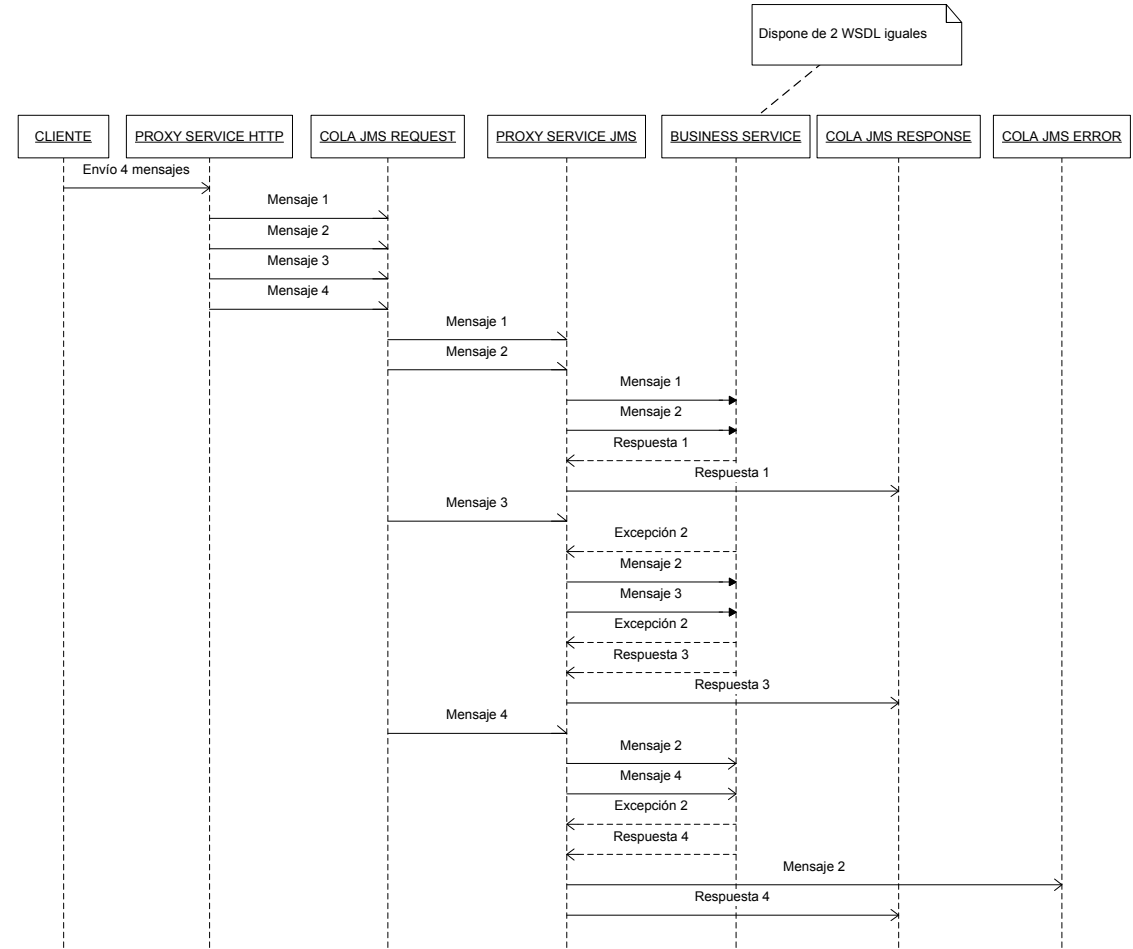
Publicar y proteger con WS-Security un WS sin protección



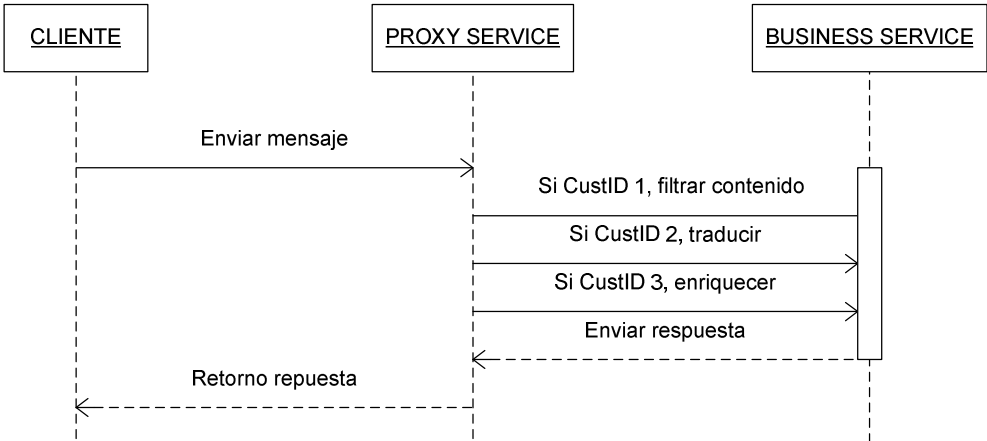
Escalabilidad en servicios síncronos



Escalabilidad en canal punto a punto

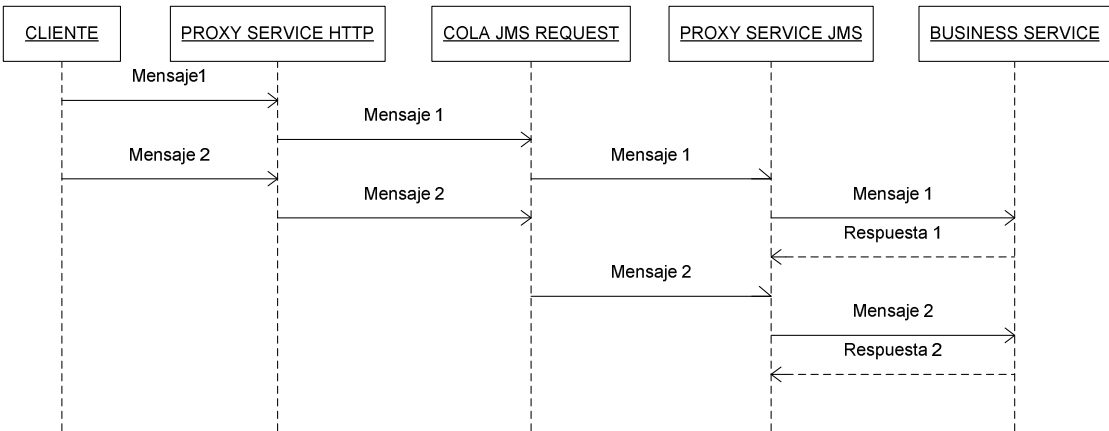


Transformación de mensajes

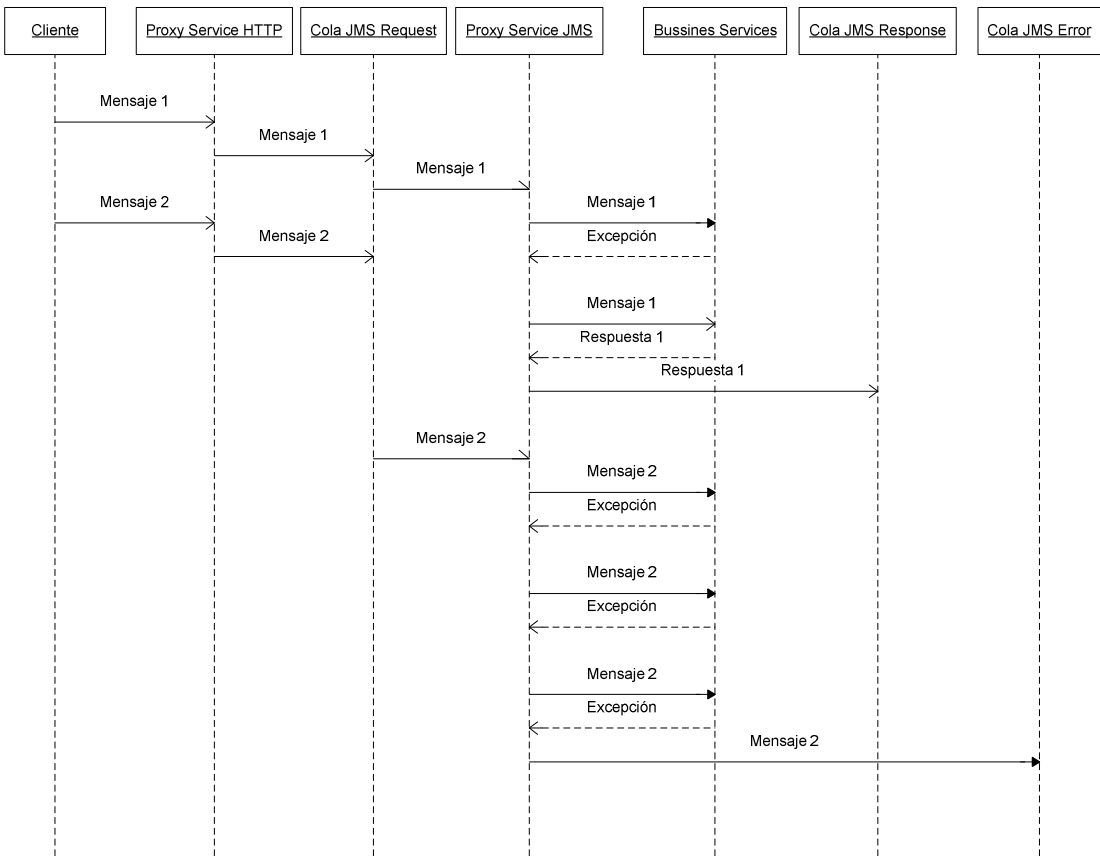


Casos de Prueba de la Sección de Procedimientos Electrónicos

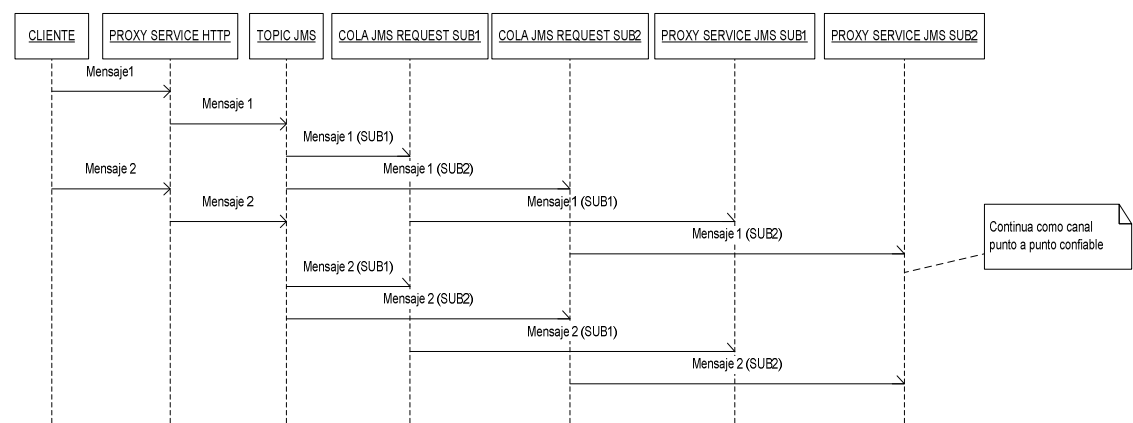
Consumidor de mensajes (Polling consumer)



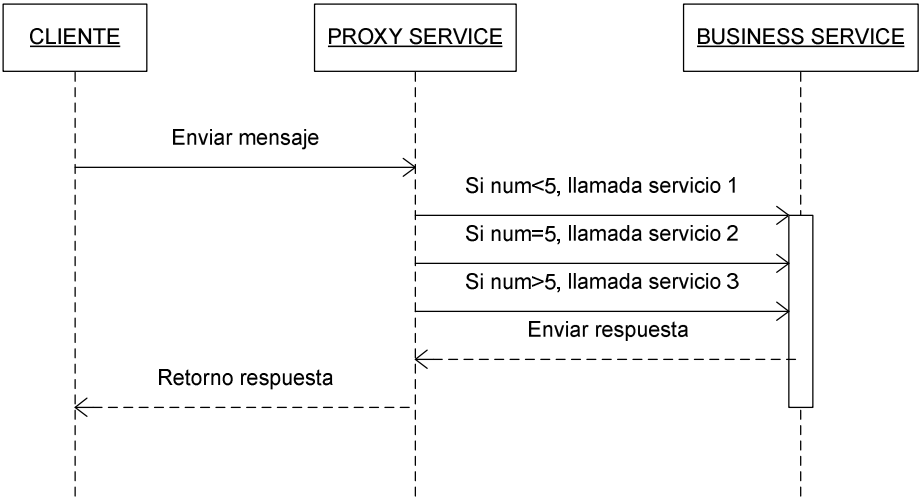
Canal punto a punto confiable



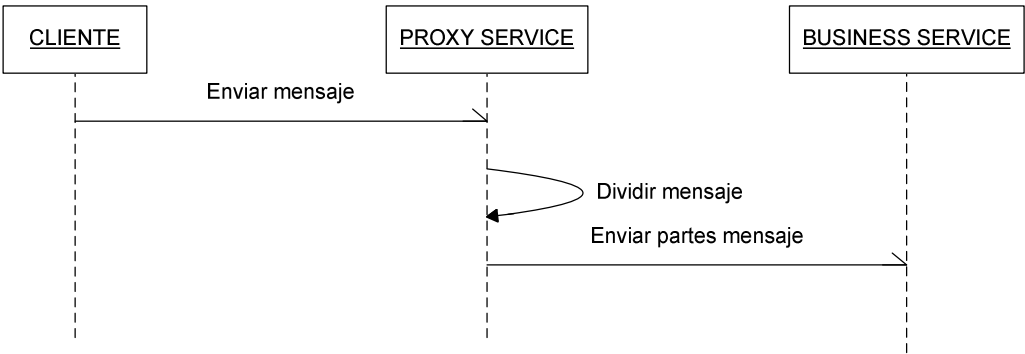
Canal de publicación / subscripción



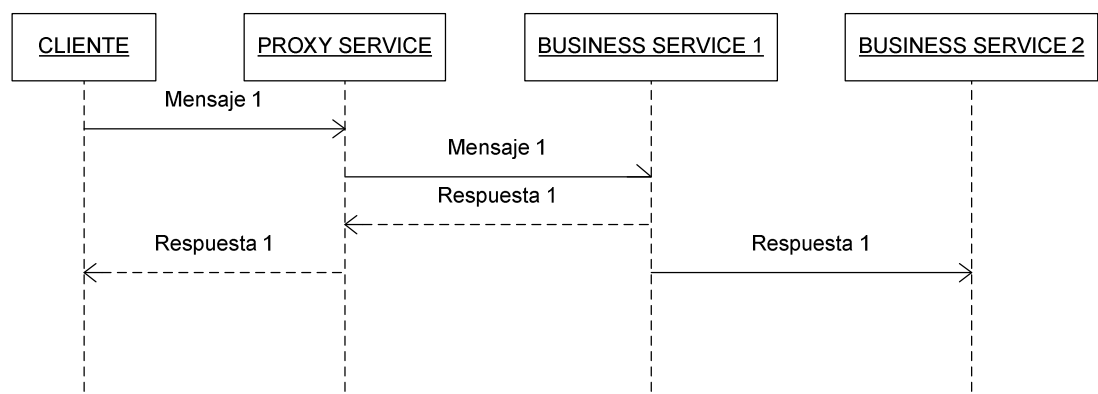
Encaminamiento basado en el contenido



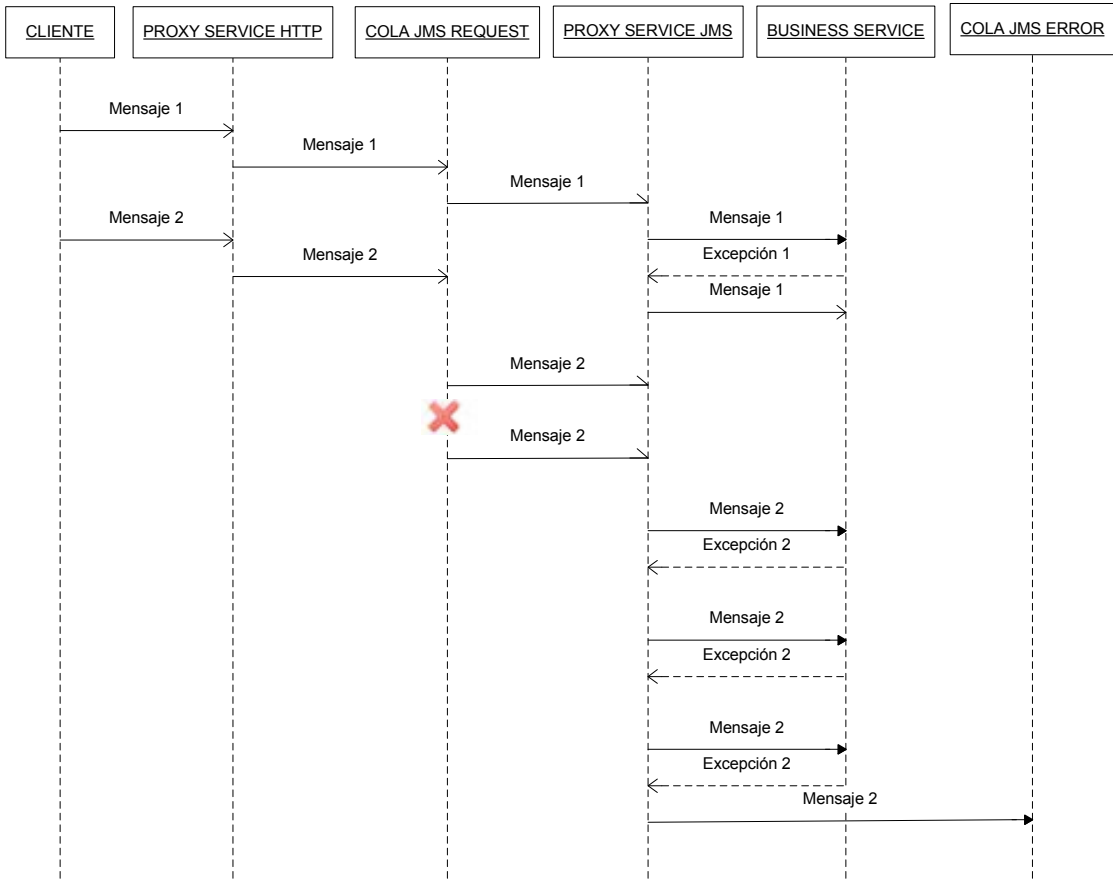
Separador



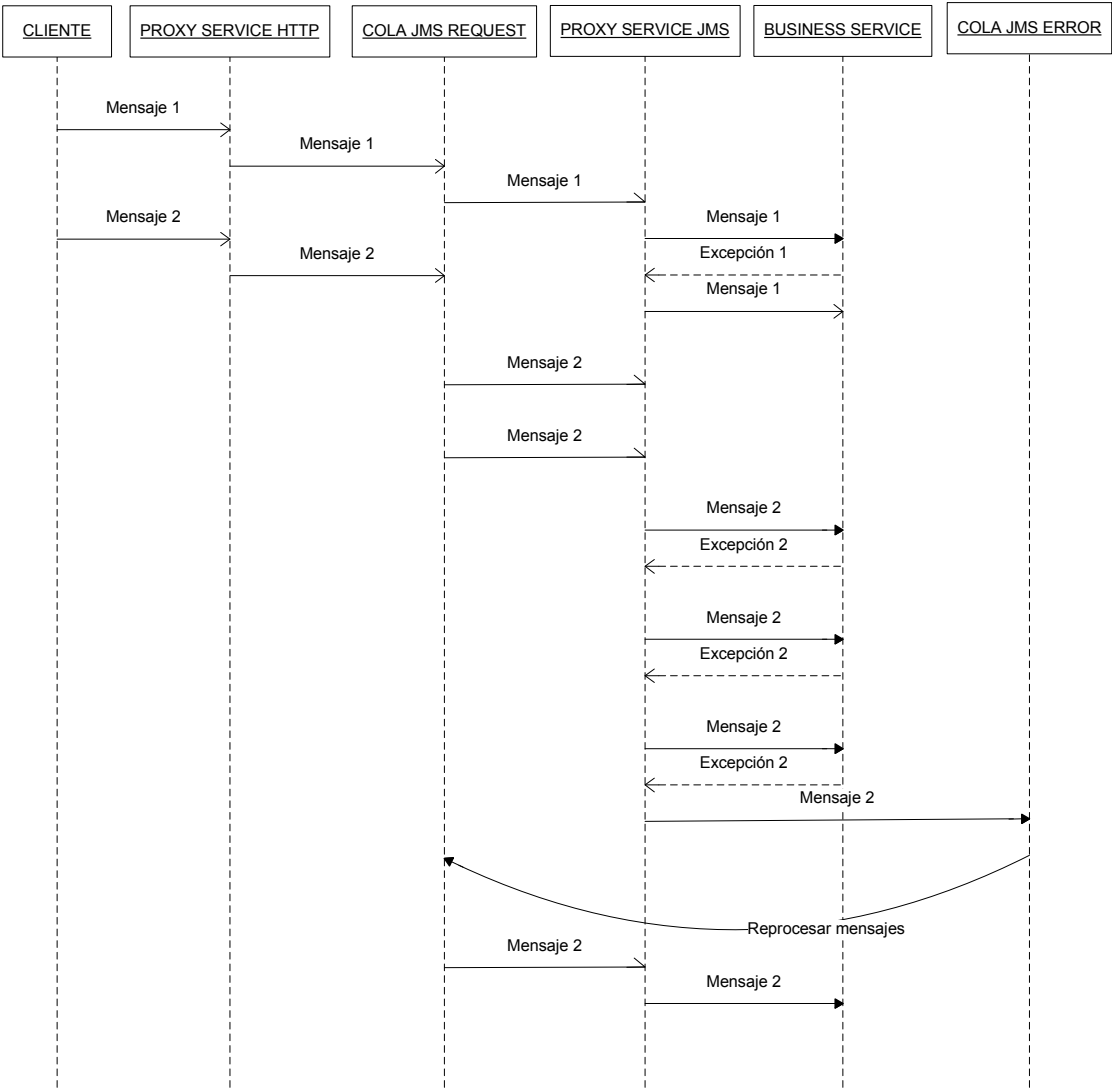
Escucha en llamada síncrona



Canal de entrega garantizada

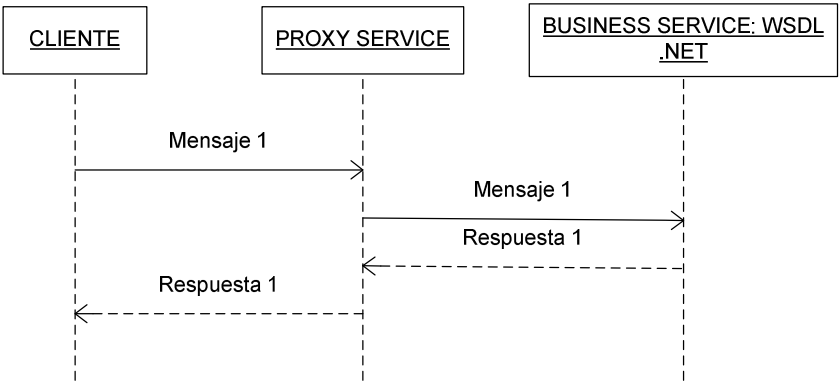


Canal de mensajes fallidos (dead letter channel)



Casos de Prueba del Grupo de Arquitectura de Desarrollo

Invocación de WS con tipos de datos propietarios de .Net desde Java



DOCUMENTO DE DESARROLLO / IMPLEMENTACIÓN

Implementación

Preparación del entorno de generación y construcción

Una de las principales características del OSB es que está construido sobre la tecnología *Oracle WebLogic Server*. Le proporciona al OSB mayor escalabilidad y fiabilidad.

Oracle WebLogic es un servidor de aplicaciones Java EE y también un servidor web HTTP desarrollado por *BEA Systems* y posteriormente adquirida por *Oracle Corporation*.

Una de las funciones más utilizadas en esta prueba de concepto ha sido la interoperabilidad entre *Oracle Service Bus* y *Weblogic JMS*:

Uno de los principales elementos para crear cada caso de prueba es tener un *Web Service*. Son *Web Services* sencillos y desarrollados en relación a cada caso, por ejemplo, para el caso de prueba “Escalabilidad en servicios síncronos” se ha creado un *Web Service* que tarda en responder 30 segundos. De esta manera, podemos ver la efectividad del balanceo de carga (en este caso).

No se han utilizado *Web Services* reales de Gobierno de Navarra porque entender el funcionamiento de cada uno de ellos alargaría mucho el desarrollo del proyecto.

Otros de los puntos que merece la pena nombrar es la forma en que las aplicaciones cliente (tanto Java como .Net) pueden consumir el WSDL publicado por el OSB.

Una vez creado el proyecto, en las propiedades del *Proxy Service*, aparece la URI de punto final a la que se debe hacer referencia, por ejemplo:

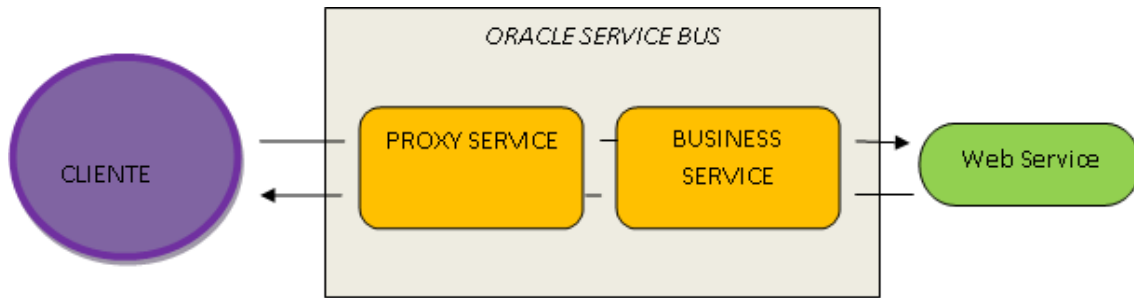
URI de Punto Final*	Formato: /someName <input data-bbox="624 1384 1348 1417" type="text" value="/CanalEntregaGarantizada/Proxy_Services/PS1"/>
----------------------------	--

Generación del código de los componentes

En el Anexo II, Documentos de Implementación, se especifica de manera detallada el desarrollo de cada uno de los casos de prueba. Además, se presenta una prueba por cada uno de ellos para comprobar su funcionamiento.

El código fuente de cada aplicación cliente como del propio proyecto OSB se adjuntan a esta memoria.

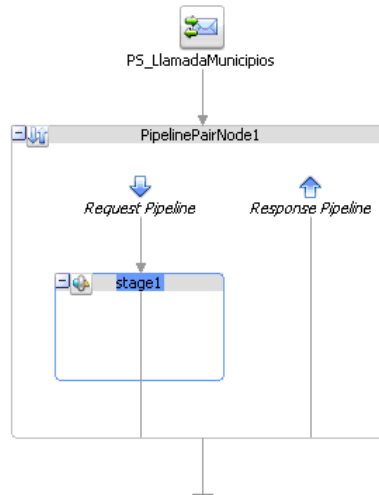
A continuación se hace una breve explicación de los elementos más importantes presentes en el desarrollo de cualquier proyecto en la consola del OSB (o en Eclipse). Entre ellos cabe destacar (Figura 32):



-Figura 32. Conceptos clave-

1. Un Business Service es la definición de un servicio de empresa que intercambia mensajes durante los procesos de negocio. Para configurar un servicio de negocio se debe especificar su interfaz (WSDL), el tipo de transporte que utiliza o los requisitos de seguridad, entre otras características.
2. Un Proxy Service es un intermediario de un *Web Service* que *Oracle Service Bus* implementa a nivel local en el servidor *WebLogic*. Los clientes de servicios intercambian mensajes con un *Proxy Service* intermediario en lugar de trabajar directamente con un servicio de negocio.
3. Un flujo de mensajes es la implementación de un *Proxy Service*. Permite configurar la lógica de la manipulación de mensajes. Esta lógica incluye actividades tales como la transformación, la publicación y presentación de informes, que se implementan como acciones individuales dentro de las etapas de un *pipeline*.

Un *pipeline* es una ruta de procesamiento que no contiene ninguna ramificación. Un *pipeline* contiene un conjunto de etapas compuestas por acciones que son las encargadas de interactuar con los mensajes. Un *pipeline* contiene dos caminos, uno para las solicitudes de servicio y otro para las respuestas.



Algunos ejemplos de acciones son:

- *Service Callout*: permite la invocación de un servicio de negocio dentro de un flujo de mensajes. La respuesta de dicho servicio se guarda en una variable la cual se puede seguir manipulando.
- *Assign*: permite asignar valores a variables, por ejemplo, el mensaje de entrada se guarda en la variable \$body por defecto, pues con la acción “Assign” podemos coger un valor del mensaje y asignarlo a una nueva variable. Además, nos permite asignar a una variable el resultado de una transformación XQuery, XPath o XSLT.
- *If Then*: permite añadir una condición, por ejemplo, según el valor que se introduce llamar a un servicio de negocio o a otro.

Esto sólo son ejemplos, OSB dispone de muchas más acciones.

En resumen, se pueden enumerar las fases de la implementación como sigue:

1. Creación del *Business Service* con las características adecuadas.
2. Creación del *Proxy Service* con las características adecuadas.
3. Implementación del flujo de mensajes del *Proxy Service*.
4. Llamada desde un cliente.

DOCUMENTO DE PRUEBAS Y VALIDACIÓN

Pruebas de validación

A continuación se realizan las pruebas de integración de cada uno de los componentes del sistema de información, una vez codificados, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

Se ha definido el entorno necesario para la realización de cada nivel de prueba, así como las verificaciones asociadas a las pruebas de integración, la coordinación y secuencia a seguir en la ejecución de las mismas y los criterios de registro y aceptación de los resultados.

Cada una de las pruebas está explicada de manera más específica en el *Anexo II: Documentos de Implementación*.

Prueba 1: Publicar y proteger con WS-Security un WS sin protección

Actividad 1:

PROCEDIMIENTO: Prueba cliente Java. Se introduce un usuario y la contraseña de acceso de directorio activo, por ejemplo usuario: NombreUsuario y contraseña:

<no mostrada por seguridad>

RESULTADO ESPERADO: acceso al servicio.

RESULTADO OBTENIDO: acceso al servicio. “Descripción pasada al servicio: Hola”.

Actividad 2:

PROCEDIMIENTO: Prueba cliente Java. Se introduce un usuario y la contraseña de acceso que no pertenezca a directorio activo, por ejemplo usuario: NombreU y contraseña: <no mostrada por seguridad>

RESULTADO ESPERADO: mensaje de error.

RESULTADO OBTENIDO: ‘General Web Service security error’.

Actividad 3:

PROCEDIMIENTO: Prueba cliente .NET. Se introduce un usuario y la contraseña de acceso de directorio activo, por ejemplo usuario: NombreUsuario y contraseña:

<no mostrada por seguridad>

RESULTADO ESPERADO: acceso al servicio.

RESULTADO OBTENIDO: acceso al servicio. “Descripción pasada al servicio: Hola”.

Actividad 4:

PROCEDIMIENTO: Prueba cliente .NET. Se introduce un usuario y la contraseña de acceso que no pertenezca a directorio activo, por ejemplo usuario: NombreU y contraseña: <no mostrada por seguridad>

RESULTADO ESPERADO: mensaje de error.

RESULTADO OBTENIDO: ‘Se recibió un error no seguro o protegido incorrectamente de la otra parte’.

Prueba 2: Escalabilidad en servicios síncronos

Actividad 1:

PROCEDIMIENTO: Paso 1. Prueba aplicación Java. Ejecutamos la aplicación la cual envía tres mensajes a la vez.

RESULTADO ESPERADO: los mensajes se envían al mismo servicio de negocio.

RESULTADO OBTENIDO: los mensajes se envían al mismo servicio de negocio.

Actividad 2:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Ejecutamos la aplicación la cual envía tres mensajes a la vez.

RESULTADO ESPERADO: los mensajes se reparten entre los diferentes servicios de negocio.

RESULTADO OBTENIDO: reparto de mensajes, 2 a un servicio y 1 a otro servicio.

Actividad 3:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Paramos los dos *Web Services* desplegados en weblogic y lanzamos la aplicación.

RESULTADO ESPERADO: mensaje de error.

RESULTADO OBTENIDO: mensaje de error.

Actividad 4:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Arrancamos los dos *Web Services* de nuevo

RESULTADO ESPERADO: continuar con el procesamiento de los mensajes.

RESULTADO OBTENIDO: los mensajes se pierden.

Actividad 5:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Paramos un único *Web Service* de los dos que utilizamos.

RESULTADO ESPERADO: los mensajes van al servicio disponible.

RESULTADO OBTENIDO: los mensajes van al servicio disponible.

Prueba 3: Escalabilidad en canal punto a punto

Actividad 1:

PROCEDIMIENTO: Paso 1. Prueba aplicación Java. Ejecutamos la aplicación la cual envía tres mensajes a la vez.

RESULTADO ESPERADO: los mensajes se envían a la misma cola JMS y se procesan a la vez.

RESULTADO OBTENIDO: los mensajes se envían a la misma cola JMS y se procesan a la vez.

Actividad 2:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Ejecutamos la aplicación la cual envía tres mensajes a la vez.

RESULTADO ESPERADO: los mensajes se reparten entre las diferentes colas JMS.

RESULTADO OBTENIDO: reparto de mensajes, 2 a un servicio y 1 a otro servicio.

Actividad 3:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Paramos los dos *Web Services* desplegados en weblogic y lanzamos la aplicación.

RESULTADO ESPERADO: mensaje de error.

RESULTADO OBTENIDO: el mensaje se guarda en la cola de error.

Actividad 4:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Arrancamos los dos *Web Services* de nuevo.

RESULTADO ESPERADO: continuar con el procesamiento de los mensajes.

RESULTADO OBTENIDO: los mensajes se mantienen en la cola de error.

Actividad 5:

PROCEDIMIENTO: Paso 2. Prueba aplicación Java. Paramos un único *Web Service* de los dos que utilizamos.

RESULTADO ESPERADO: los mensajes van al servicio disponible.

RESULTADO OBTENIDO: los mensajes van al servicio disponible.

Prueba 4: Transformación de mensajes

Actividad 1:

PROCEDIMIENTO: Paso 1. Ejecutamos la consola del OSB. Introducimos en custID el valor 1. Introducimos los valores de cada campo.

RESULTADO ESPERADO: Valor campo 'description'.

RESULTADO OBTENIDO: Valor campo 'description'.

Actividad 2:

PROCEDIMIENTO: Paso 2. Ejecutamos la consola del OSB. Introducimos en custID el valor 2. Introducimos los valores de cada campo.

RESULTADO ESPERADO: Visualización de los campos 'description' y 'price' juntos y del campo 'departament' en mayúsculas.

RESULTADO OBTENIDO: Visualización de los campos 'description' y 'price' juntos y del campo "departament" en mayúsculas.

Actividad 3:

PROCEDIMIENTO: Paso 3. Ejecutamos la consola del OSB. Introducimos en custID el valor 3. Introducimos los valores de cada campo.

RESULTADO ESPERADO: enriquecimiento del mensaje.

RESULTADO OBTENIDO: se han añadido dos campos nuevos al mensaje, nombre y apellidos.

Prueba 5: Consumidor de mensajes (Polling consumer)

Actividad 1:

PROCEDIMIENTO: Realizamos varias llamadas desde la consola de prueba del OSB.

RESULTADO ESPERADO: los mensajes llegan a la cola JMS de respuesta.

RESULTADO OBTENIDO: los mensajes llegan a la cola JMS "response".

Prueba 6: Canal punto a punto confiable

Actividad 1:

PROCEDIMIENTO: Paso 2. Ejecutamos varias veces PS1 desde la consola de prueba del OSB.

RESULTADO ESPERADO: Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación.

RESULTADO OBTENIDO: Obtenemos una respuesta inmediata. En este ejemplo, se ha ejecutado el PS1 6 veces, de las cuales 4 han tenido éxito y 2 han fallado.

Actividad 2:

PROCEDIMIENTO: Paso 3. Ejecutamos varias veces PS1 desde la consola de prueba del OSB.

RESULTADO ESPERADO: Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación. Las peticiones se reparten entre los distintos servicios de negocio.

RESULTADO OBTENIDO: Obtenemos una respuesta inmediata y las peticiones se reparten entre los distintos servicios de negocio, en concreto 2 a un servicio y 1 a otro servicio.

Actividad 3:

PROCEDIMIENTO: Paso 3. Paramos los dos *Web Services* desplegados en weblogic y ejecutamos varias veces PS1 desde la consola de prueba del OSB.

RESULTADO ESPERADO: mensaje de error.

RESULTADO OBTENIDO: los mensajes se guardan en la cola de error.

Actividad 4:

PROCEDIMIENTO: Paso 3. Arrancamos los dos *Web Services* de nuevo.

RESULTADO ESPERADO: continuar con el procesamiento de los mensajes.

RESULTADO OBTENIDO: los mensajes se mantienen en la cola de error.

Actividad 5:

PROCEDIMIENTO: Paso 3. Paramos un único *Web Service* de los dos que utilizamos y ejecutamos varias veces PS1 desde la consola de prueba del OSB.

RESULTADO ESPERADO: los mensajes van al servicio disponible.

RESULTADO OBTENIDO: los mensajes van al servicio disponible.

Prueba 7: Canal de publicación / subscripción

Actividad 1:

PROCEDIMIENTO: Paso 2. Ejecutamos varias veces PSPublicador desde la consola de prueba del OSB.

RESULTADO ESPERADO: Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación. Los mensajes son consumidos por todos los subscriptores.

RESULTADO OBTENIDO: Obtenemos una respuesta inmediata. En este ejemplo se han realizado 6 llamadas y cada subscritor a recibido los 6 mensajes.

Prueba 8: Encaminamiento basado en el contenido

Actividad 1:

PROCEDIMIENTO: Introducimos, Nombre: almudena, Apellido: ochoa, Número: 3.

RESULTADO ESPERADO: Llamada servicio 1

RESULTADO OBTENIDO: Llamada servicio 1

Actividad 2:

PROCEDIMIENTO: Introducimos, Nombre: almudena, Apellido: ochoa, Número: 5.

RESULTADO ESPERADO: Llamada servicio 2

RESULTADO OBTENIDO: Llamada servicio 2

Actividad 3:

PROCEDIMIENTO: Introducimos, Nombre: almudena, Apellido: ochoa, Número: 7.

RESULTADO ESPERADO: Llamada servicio 3

RESULTADO OBTENIDO: Llamada servicio 3

Prueba 9: Separador

Actividad 1:

PROCEDIMIENTO: ejecutamos desde la consola de prueba del OSB. Introducimos dos peticiones, es decir, dos ítems.

RESULTADO ESPERADO: Tratamiento de las peticiones por separado.

RESULTADO OBTENIDO: cada petición se muestra el campo descripción. Tratamiento de las peticiones por separado.

Prueba 10: Canal de entrega garantizada

Actividad 1:

PROCEDIMIENTO: Si utilizamos un *Web Service one-way*. Ejecutamos varias veces PS1 desde la consola de prueba del OSB. Paramos el servidor de aplicaciones (weblogic). Lo volvemos a arrancar.

RESULTADO ESPERADO: Los mensajes no se pierden, se continúan procesando.

RESULTADO OBTENIDO: Los mensajes no se pierden, se continúan procesando.

Actividad 2:

PROCEDIMIENTO: Si utilizamos un *Web Service request-response*. Ejecutamos varias veces PS1 desde la consola de prueba del OSB. Paramos el servidor de aplicaciones (weblogic). Lo volvemos a arrancar.

RESULTADO ESPERADO: Los mensajes no se pierden, se continúan procesando.

RESULTADO OBTENIDO: Los mensajes se pierden. Se muestra de manera continua el siguiente error:

```
<Warning> <WlsBTransports> <BEA-381542> <JmsAsyncResponseMDB.onMessage()
ha recibido un mensaje con un identificador de mensaje desconocido
bb4a176d0000012dc66feaaeffff804a; marcando la transacciÃ³n para rollback...>
```

Prueba 11: Canal de mensajes fallidos (dead letter channel)

Actividad 1:

PROCEDIMIENTO: Parar el wsdl en weblogic. Ejecutar PS1 tres veces. Los mensajes van a la cola de error. Reiniciar el wsdl. Mover los mensajes a la cola PS2LentoPeticion.

RESULTADO ESPERADO: Los mensajes vuelven a ser procesados.

RESULTADO OBTENIDO: Los mensajes vuelven a ser procesados.

Prueba 12: Invocación de WS con tipos de datos propietarios de .Net desde Java.

Actividad 1:

PROCEDIMIENTO: Introducimos un entero y un string cualquiera.

RESULTADO ESPERADO: Devuelve los datos del DataSet.

RESULTADO OBTENIDO: Devuelve los datos del DataSet.

DOCUMENTO DE CONCLUSIONES

Presentación de informes a los usuarios finales

A continuación se presentan los informes con los resultados obtenidos en la prueba de concepto. Para cada caso de prueba se ha realizado un informe en el que se presentan las principales técnicas utilizadas y las conclusiones según los resultados obtenidos en el desarrollo de los mismos.

Estos resultados son los que se exponen a los ‘solicitantes’ en la presentación final.

Publicar y proteger con WS-Security un WS sin protección⁵
TÉCNICAS UTILIZADAS
<p>Existen tres maneras de añadir políticas a nivel de mensaje:</p> <ul style="list-style-type: none"> • Enlaces de Política de OWSM: Políticas predefinidas por <i>Oracle Web Service Manager</i>, por ejemplo: <code>oracle/wss_username_token_service_policy</code>. • Política basada en WSDL (Políticas WSDL 9.2): utilizar las políticas que se encuentran definidas en los propios WSDL. • Enlaces de políticas Personalizadas (Políticas WSDL 9.2): nos permite añadir políticas creadas por nosotros o utilizar las predefinidas como por ejemplo: <p>Auth.xml: Esta política requiere que el cliente incluya un token de autenticación en el mensaje</p> <p>Encrypt.xml: Esta política requiere que el cliente cifre el cuerpo SOAP con 3DES-CBC.</p> <p>Sign.xml: Esta política requiere que los clientes firmen el cuerpo SOAP.</p>
PROBLEMAS ENCONTRADOS O NO RESUELTOS
No se han encontrado.
CONCLUSIONES
<p>Interoperabilidad de las políticas predefinidas de OWSM 11g y Java</p> <p>Java es interoperable con cualquiera de las políticas del estándar <i>WS-Security</i>. En este caso se ha utilizado <code>oracle/wss_username_token_service_policy</code>. Esta política de seguridad no es 'digest', es decir, que en el mensaje SOAP el usuario y password van en claro (sin encriptar). Java es interoperable con políticas 'digest' como 'no digest'. Esta política en concreto sólo es recomendable utilizarla en pruebas.</p> <p>Interoperabilidad de las políticas predefinidas de OWSM 11g y Microsoft Windows Communication Foundation (WCF)/.NET 3.5</p> <p>Si queremos proteger a nivel de mensaje con Username Token y que sea interoperable con Microsoft WCF/.Net 3.5 es necesario que las políticas OWSM sigan el estándar <i>WS-Security</i> 1.1.</p> <p><u>En la consola de <i>Oracle Service Bus</i> 11g</u></p> <ul style="list-style-type: none"> • Añadimos la política en el <i>Proxy Service</i>: <code>oracle/wss11_username_token_with_message_protection_service_policy</code>

⁵ http://download.oracle.com/docs/cd/E12839_01/web.1111/b32511/interop.htm#BABCJGJI

- Exportar el certificado X.509 del keystore (a nuestra máquina).

En el cliente .Net

Es necesario modificar el archivo app.config (web.config en caso de que sea aplicación web).

Formato archivo app.config (se debe modificar las partes en negrita según cada caso):

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.serviceModel>
    <behaviors>
      <endpointBehaviors>
        <behavior name="secureBehaviour">
          <clientCredentials>
            <serviceCertificate>
              <defaultCertificate findValue="<certificate_cn>"
                storeLocation="CurrentUser" storeName="My"
                x509FindType="FindBySubjectName"/>
            </serviceCertificate>
          </clientCredentials>
        </behavior>
      </endpointBehaviors>
    </behaviors>
    <bindings>
      <customBinding>
        <binding name="HelloWorldSoapHttp">
          <security defaultAlgorithmSuite="Basic128"
            authenticationMode="UserNameForCertificate"
            requireDerivedKeys="false" securityHeaderLayout="Lax"
            includeTimestamp="true"
            keyEntropyMode="CombinedEntropy"
            messageProtectionOrder="SignBeforeEncrypt"
            messageSecurityVersion=
"WSSecurity11WSTrustFebruary2005WSSecureConversationFebruary2005WSSecurityP
olicy11BasicSecurityProfile10"
            requireSignatureConfirmation="true">
            <localClientSettings cacheCookies="true" detectReplays="false"
              replayCacheSize="900000" maxClockSkew="00:05:00"
              maxCookieCachingTime="Infinite" replayWindow="00:05:00"
              sessionKeyRenewalInterval="10:00:00" sessionKeyRolloverInterval="00:05:00"
              reconnectTransportOnFailure="true" timestampValidityDuration="00:05:00"
              cookieRenewalThresholdPercentage="60" />
            <localServiceSettings detectReplays="true" issuedCookieLifetime="10:00:00"
              maxStatefulNegotiations="128" replayCacheSize="900000"
              maxClockSkew="00:05:00" negotiationTimeout="00:01:00"
              replayWindow="00:05:00" inactivityTimeout="00:02:00"
              sessionKeyRenewalInterval="15:00:00" sessionKeyRolloverInterval="00:05:00"
              reconnectTransportOnFailure="true" maxPendingSessions="128"
```

```

    maxCachedCookies="1000" timestampValidityDuration="00:05:00" />
<secureConversationBootstrap /></security>
<textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16"
messageVersion="Soap11" writeEncoding="utf-8">
    <readerQuotas maxDepth="32" maxStringContentLength="8192"
    maxArrayLength="16384" maxBytesPerRead="4096"
    maxNameTableCharCount="16384" />
</textMessageEncoding>
<HttpTransport manualAddressing="false" maxBufferPoolSize="524288"
maxReceivedMessageSize="65536" allowCookies="false"
authenticationScheme="Anonymous" bypassProxyOnLocal="false"
hostNameComparisonMode="StrongWildcard" keepAliveEnabled="true"
maxBufferSize="65536" proxyAuthenticationScheme="Anonymous"
realm="" transferMode="Buffered" unsafeConnectionNtlmAuthentication="false"
useDefaultWebProxy="true" />
</binding>
</customBinding>
</bindings>
<client>
    <endpoint address="<endpoint_url>"
    binding="customBinding"
    bindingConfiguration="<mywebservice>SoapHttp"
    contract="<mywebservice>"
    name="<mywebservice>Port"
    behaviorConfiguration="secureBehaviour" >
        <identity>
            <dns value="<certificate_cn>" />
        </identity>
    </endpoint>
</client>
</system.serviceModel>
</configuration>

```

-Tabla 19. Informe: Publicar y proteger con WS-Security un WS sin protección-

Escalabilidad en servicios síncronos
TÉCNICAS UTILIZADAS
<p>Si se tiene un servicio que espera recibir un uso intensivo de mensajes es necesario crear varias instancias del servicio y distribuir las solicitudes a través de ellas a fin de compartir la carga de trabajo. Esto se denomina balanceo de carga. <i>Oracle Service Bus</i> posee los siguientes algoritmos de carga:</p> <ul style="list-style-type: none"> • Asignación en rueda (Round Robin) - la primera solicitud va al primer extremo del servicio, la segunda petición va al segundo extremo del servicio y así sucesivamente hasta que todos los extremos se utilicen y luego se repite la secuencia. • Al azar - cada solicitud se distribuye a cualquiera de los extremos de su servicio. • Aleatoria ponderada – a cada extremo de servicio se le da una ponderación y las solicitudes de servicio se distribuyen aleatoriamente entre los extremos de servicio (según la ponderación que se ha definido). • Ninguna. <p>A parte del balanceo de carga, el bus permite:</p> <ul style="list-style-type: none"> • Gestionar caudal de cada una de las instancias: permite indicar el número máximo de peticiones que puede aceptar cada uno de los extremos. • Encolar peticiones: Permite indicar el número máximo de mensajes en espera en la cola. Si los extremos se encuentran ocupados los mensajes se encolan hasta que alguno de los extremos quede libre. • Reintentos (posibilidad de reintentos): permite indicar el número de reintentos en caso de que al acceder a uno de los extremos éste se encuentre ocupado.
PROBLEMAS ENCONTRADOS O NO RESUELTOS
No se han encontrado.
CONCLUSIONES
A continuación se presenta una tabla comparativa entre la utilización de una o dos instancias de un <i>Web Service</i> .

	1 URL	2 URLs
9 Llamadas	30 segundos	30 segundos
12 Llamadas	60 segundos	30 segundos
21 Llamadas	90 segundos	60 segundos
30 Llamadas	120 segundos	90 segundos

El número máximo de peticiones tratadas por cada *Web Service* se ha definido en 9. Como se puede observar en los resultados expuestos en la tabla, al utilizar dos instancias del mismo *Web Service* no se reduce a la mitad el tiempo en procesar las llamadas pero disminuye considerablemente el tiempo de ejecución. El OSB permite gestionar el caudal de cada una de las instancias y permite encolar las peticiones que se encuentran en espera.

-Tabla 20. Informe: Escalabilidad en servicios síncronos-

Escalabilidad en canal punto a punto		
TÉCNICAS UTILIZADAS		
Igual que en Escalabilidad de servicios síncronos.		
PROBLEMAS ENCONTRADOS O NO RESUELTOS		
No se han encontrado.		
CONCLUSIONES		
A continuación se presenta una tabla comparativa entre la utilización de una o dos instancias de un <i>Web Services</i> .		
	1 URL	2 URLs
9 Llamadas	30 segundos	30 segundos
12 Llamadas	60 segundos	30 segundos
21 Llamadas	90 segundos	60 segundos
30 Llamadas	120 segundos	90 segundos
<p>El número máximo de peticiones tratadas por cada <i>Web Service</i> se ha definido en 9. Como se puede observar en los resultados expuestos en la tabla, al utilizar dos instancias del mismo <i>Web Service</i> no se reduce a la mitad el tiempo en procesar las llamadas pero disminuye considerablemente el tiempo de ejecución.</p> <p>El OSB permite gestionar el caudal de cada una de las instancias y permite encolar las peticiones que se encuentran en espera.</p> <p>La ventaja que nos proporciona utilizar escalabilidad en canal punto a punto es que además del balanceo de carga nos permite obtener un canal confiable.</p>		

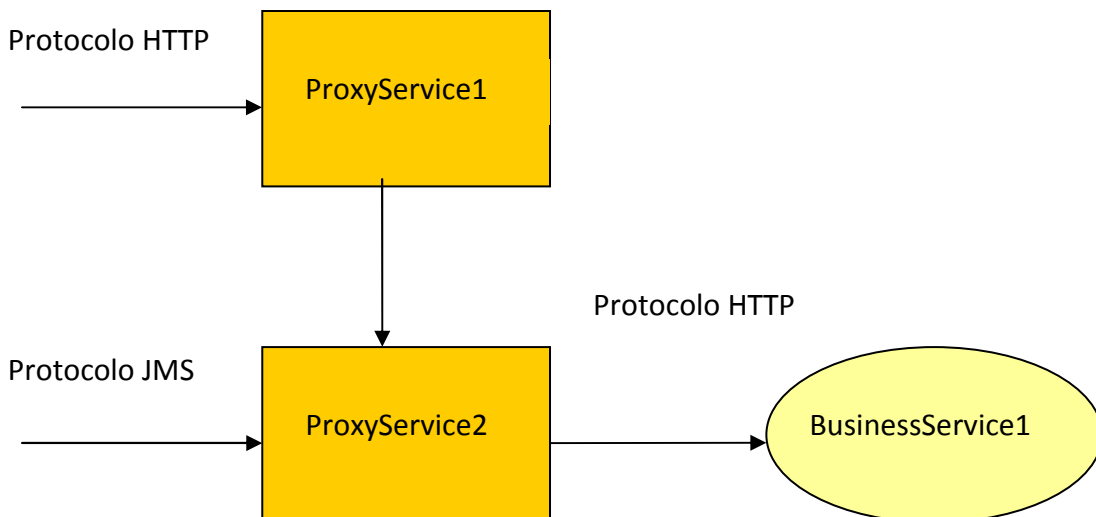
-Tabla 21. Informe: Escalabilidad en canal punto a punto -

Transformación de mensajes
TÉCNICAS UTILIZADAS
<p>1. <u>Tecnologías para la transformación de mensajes.</u></p> <p>OSB agrega una capa de transformación de mensajes que es útil para:</p> <ul style="list-style-type: none"> • Mapear distintos datos entre el origen y el destino. • Manipular el contenido del mensaje. <p>OSB proporciona apoyo a la transformación de mensajes usando tres tecnologías estándares: XSLT, XPath y XQuery.</p> <p>Se puede utilizar XSLT para transformar datos XML de un esquema a otro. Del mismo modo, se puede utilizar XQuery y XPath para representar esquemas de transformación en documentos XML a XML, XML a no-XML, y no-XML a XML.</p> <p>Además, OSB apoya el uso de <i>Message Format Language</i>(MFL) para transformaciones en formatos no XML, como valores separados por comas (ficheros CSV), libros de copia COBOL, documentos de Intercambio Electrónico de Datos (EDI, <i>Electronic Data Interchange</i>), y así sucesivamente.</p> <p>2. <u>Split Join</u></p> <p>Split-Join permite al servicio Proxy realizar múltiples llamadas en serie o en paralelo y luego devolver los resultados en una sola declaración. En este ejemplo queremos llamar de forma paralela a dos operaciones de un mismo servicio de negocio y que a su vez lo una al contenido que hemos introducido inicialmente.</p>
PROBLEMAS ENCONTRADOS O NO RESUELTOS
<p>No se puede implementar una aplicación cliente para este caso ya que cada uno de los servicios de negocio devuelve un mensaje diferente al esperado por el Proxy. Para poder realizar un cliente es necesario separar cada paso (Paso1, Paso2, Paso3) en diferentes proyectos y que cada servicio de negocio devuelva el formato de mensaje esperado por el Proxy.</p>
CONCLUSIONES
<p><i>Oracle Service Bus</i> permite crear de manera sencilla las transformaciones de mensajes a través de transformaciones XQuery. Al contrario que en la consola del OSB, Eclipse dispone de un editor gráfico que permite crear las transformaciones de una manera más sencilla.</p> <p>Una de las ventajas de utilizar Split Join es que disminuye el tiempo de servicio. Si por</p>

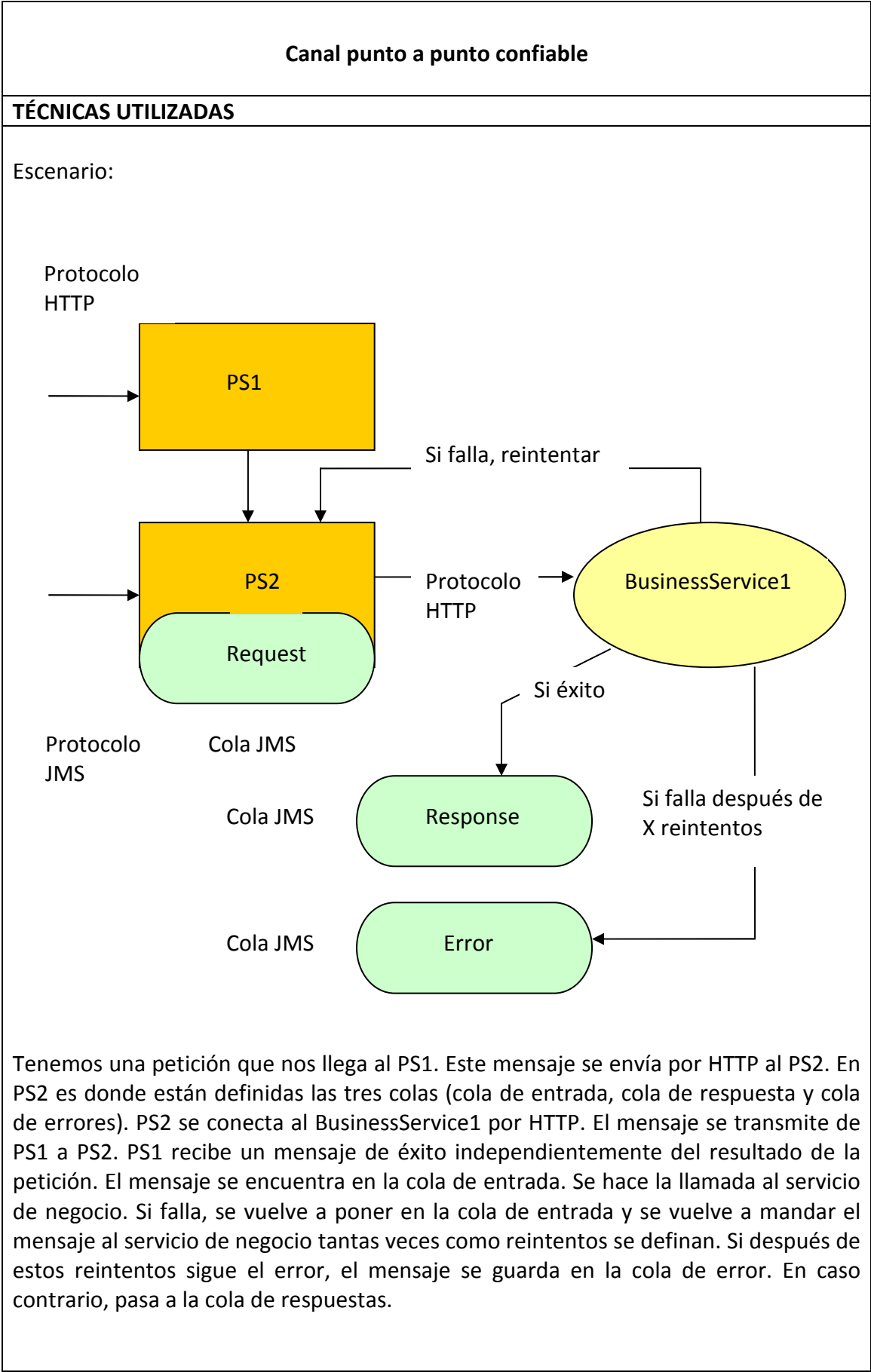
ejemplo cada llamada al servicio tarda 5 segundos en terminar, el tiempo total de servicio del Proxy serían 10 segundos (2 llamadas secuenciales). En cambio, al hacer la llamada paralela, el tiempo de servicio se reduciría a la mitad más o menos, 5 segundos.

Utilizar las transformaciones nos permite adaptar contratos WSDL, es decir, podemos adaptar los datos devueltos por un servicio al contrato que necesitemos.

-Tabla 22. Informe: Transformación de mensajes-

Consumidor de mensajes (Polling consumer)	
TÉCNICAS UTILIZADAS	
 <pre> graph TD HTTP1[Protocolo HTTP] --> ProxyService1[ProxyService1] ProxyService1 -- "Protocolo HTTP" --> ProxyService2[ProxyService2] JMS[Protocolo JMS] --> ProxyService2 ProxyService2 -- "Protocolo HTTP" --> BusinessService1([BusinessService1]) </pre> <p>Tenemos una petición que nos llega al ProxyService1. Este Proxy envía el mensaje por HTTP al ProxyService2. En el ProxyService2 es donde está definido el protocolo JMS, es decir, la cola. El ProxyService2 envía el mensaje al BusinessService1 por HTTP.</p>	
PROBLEMAS ENCONTRADOS O NO RESUELTOS	
No se han encontrado.	
CONCLUSIONES	
<p>Según el libro “The Definitive Guide to SOA. <i>Oracle Service Bus</i>” resuelve este problema implementando un MDB (message-driven bean) como consumidor de mensajes.</p> <p>Se ha llegado a la conclusión de que no es necesario crear un consumidor de mensajes para ser intermediario entre la cola de mensajes y el servicio de negocio. En <i>Oracle Service Bus</i> es la propia cola JMS la que actúa como consumidor de los mensajes ya que dispone de un MDB propio implementado.</p> <p>Los WSDL 1.1 soportan dos tipos de transmisión: one-way y request-response. En este caso, como no se ha indicado que se necesita respuestas, es necesario crear un WSDL unidireccional, es decir, de un único sentido.</p> <p>En caso de que se tenga un wsdl de petición / respuesta, se debe indicar que se necesita respuesta y crear la cola JMS correspondiente de respuesta ya que las colas JMS son unidireccionales.</p>	

-Tabla 23. Informe: Consumidor de mensajes (Polling consumer)-



PROBLEMAS ENCONTRADOS O NO RESUELTOS
No se han encontrado.
CONCLUSIONES
<p>En este caso es necesario crear una transacción global para evitar la pérdida de los mensajes.</p> <p>Si se produce un error, el mensaje no se pierde sino que se guarda en una cola JMS. Permite realizar reintentos por si el error se ha dado en un momento puntual.</p> <p>Guardar los mensajes no procesados en una cola de error aporta confiabilidad al sistema.</p>

-Tabla 24. Informe: Canal punto a punto confiable-

Canal de publicación / subscripción

TÉCNICAS UTILIZADAS

Existen dos tipos de colas en *Java Message Service* o JMS; *Topic* y *Queue*:

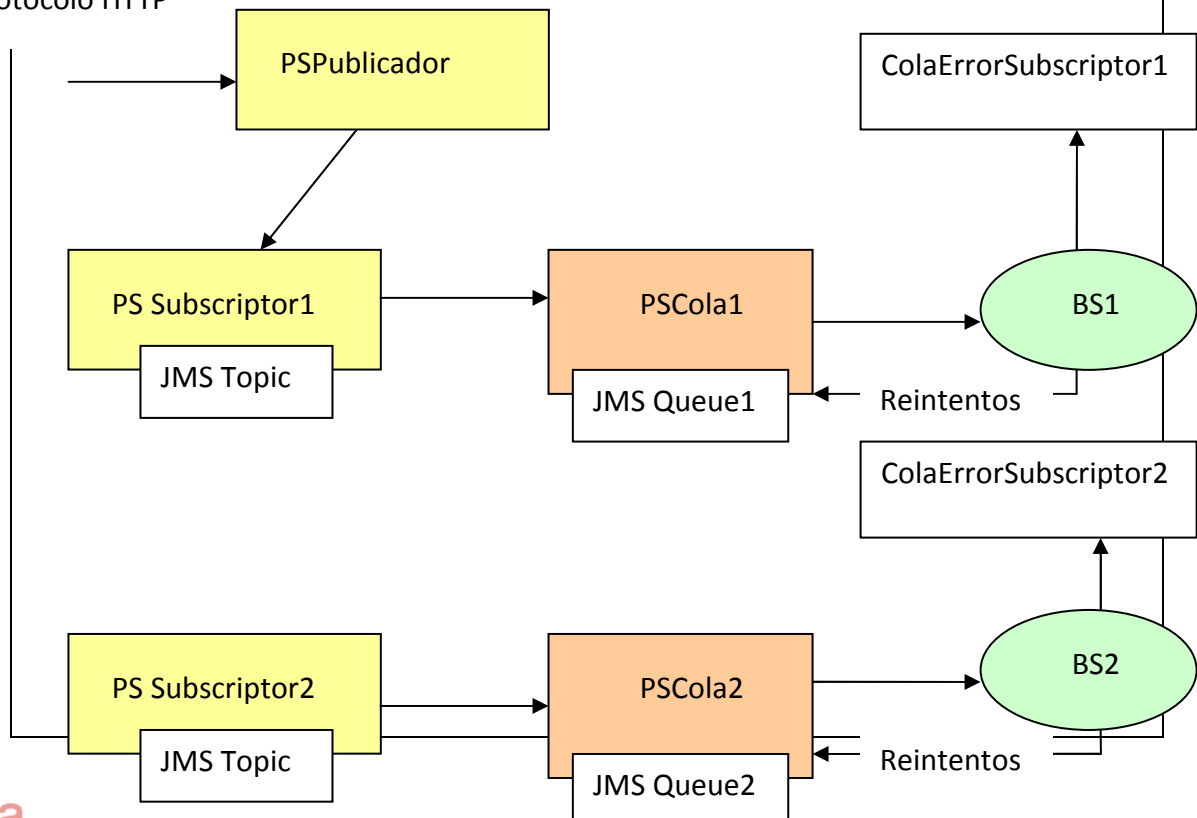
En las colas tipo *Queue*, existen uno o varios publicadores, y un consumidor. Los publicadores van dejando sus mensajes en la cola, y son tomados en orden por el consumidor. Si el consumidor no está disponible, la cola va guardando los mensajes, de manera que el consumidor pueda retomar su procesamiento cuando vuelva a estar disponible.

En las colas tipo *Topic*, siguen el modelo "publicador/subscriptor". Uno o varios consumidores se "suscriben" al *Topic*, y van recibiendo los mensajes que se publican. Cuando se desconectan dejan de recibir esos mensajes, y los pierden. De esta forma, podemos decir que:

- Las colas tipo *Topic* se utilizan cuando la información es "time sensitive", por ejemplo en las quotas de stocks...
- También utilizaremos las colas *Topic* cuando la información se envía a una audiencia amplia.
- En cambio utilizaremos colas de tipo *Queue* cuando realicemos transacciones.

Escenario:

Protocolo HTTP



<p>Tenemos una petición que nos llega a PSPublicador. Este Proxy se conecta por HTTP al PSSubscriptor2. En PSSubscriptor2 es donde está definido el <i>Topic</i>. Este <i>Topic</i> tiene que ser el mismo en todos los subscriptores. PSPublicador publica el mensaje en el <i>Topic</i> y todos los subscriptores tienen acceso a él. Una vez que cada subscriptor tiene el mensaje, lo envía a su cola de peticiones y lo trata independientemente al resto de subscriptores.</p>
<p>PROBLEMAS ENCONTRADOS O NO RESUELTOS</p>
<p>No se han encontrado.</p>
<p>CONCLUSIONES</p>
<p>En este caso es necesario crear una transacción global para evitar la pérdida de los mensajes.</p> <p>Si se produce un error, el mensaje no se pierde sino que se guarda en una cola JMS. Permite realizar reintentos por si el error se ha dado en un momento puntual.</p> <p>Permite publicar un mensaje en varios subscriptores además de añadir confiabilidad al sistema.</p>

-Tabla 25. Informe: Canal de publicación / subscripción-

Separador

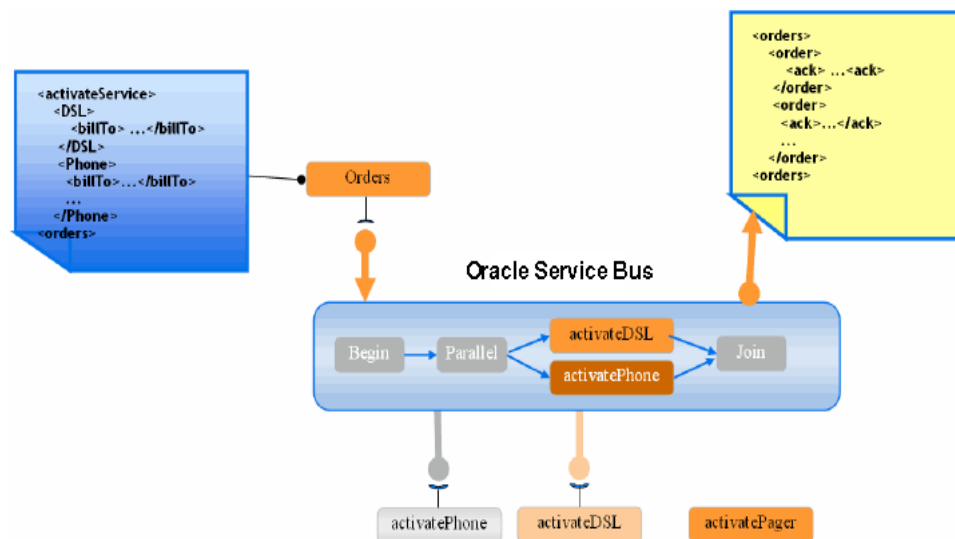
TÉCNICAS UTILIZADAS

Split Join: permite dividir una carga útil de servicio, como por ejemplo una orden, en mensajes individuales para el procesamiento concurrente. El procesamiento concurrente, en comparación con el tratamiento secuencial, mejora notablemente el rendimiento del servicio. Split-Join logra esta tarea mediante el fraccionamiento de una carga útil del mensaje de entrada en sub-mensajes (división), enrutándolos al mismo tiempo a sus destinos, y la agregación de las respuestas en un mensaje de respuesta global (unirse). Este proceso de división de la carga útil y la agregación de respuesta se llama Split-Join.

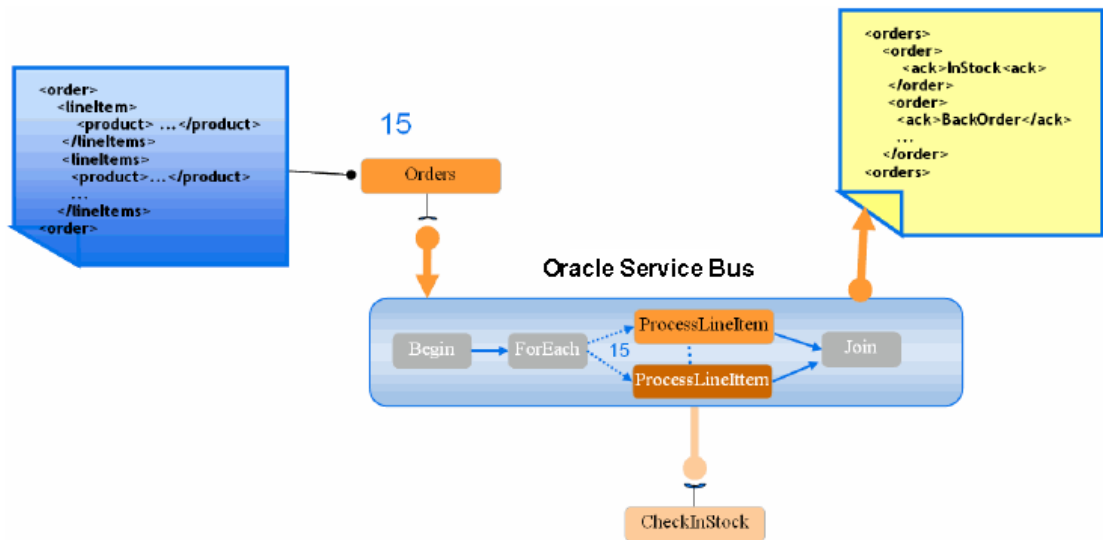
Con Split-Join, varios mensajes se procesan simultáneamente, lo que reduce la carga sobre el sistema de respuesta y mejora en gran medida los tiempos de respuesta.

Hay dos modelos de Split-Join:

- Split-Join estático: permite dividir solicitudes de servicios conocidos. Cada una de las solicitudes se procesa de manera paralela y se unen las respuestas en una sola respuesta. Permite la llamada paralela de varios servicios.



- Split-Join dinámico: utiliza la lógica condicional para determinar el número de ramas que debe crear. Todas las solicitudes se gestionan de forma simultánea, y las respuestas se agrupan en una sola respuesta.

	
PROBLEMAS ENCONTRADOS O NO RESUELTOS	
No se han encontrado.	
CONCLUSIONES	
<p>En este caso se ha utilizado Split Join dinámico ya que no se conoce el número de ítems que se van a recibir. Permite separar cada uno de los ítems y trata cada uno de manera separada y paralelamente.</p> <p>Mejora los tiempos de respuesta.</p>	

-Tabla 26. Informe: Separador-

Escucha en llamada síncrona
TÉCNICAS UTILIZADAS
<p><u>Definición de alertas</u></p> <p>Uno de los aspectos de intermediación de servicios que soporta OSB es la definición y gestión de alertas de servicio. Entendiendo que una alerta es una señal que el sistema generará a lo largo de la ejecución de un servicio, es posible definir distintos tipos de alertas desde la consola de administración de OSB: alertas sobre condiciones de error y excepción en la ejecución de los servicios, alertas basadas en SLAs y alertas configurables. Estas últimas son aquellas definidas en concepto de calidad y requerimientos del servicio como pueden ser: contenido erróneo de los mensajes de entrada/salida del mensaje, usuarios en listas negras, invocación denegada de servicios en función del origen del cliente, del destino seleccionado o de la franja horaria, etc.) OSB contempla la gestión de dichas alertas sobre diferentes mecanismos: cuadros de mando en la propia consola de administración del entorno y / o generación de mensajes JMS, correo electrónico y SNMP.</p>
PROBLEMAS ENCONTRADOS O NO RESUELTOS
No se han encontrado.
CONCLUSIONES
Permite realizar operaciones totalmente transparentes para la aplicación cliente y el servicio de negocio.

-Tabla 27. Informe: Escucha en llamada síncrona-

Canal de entrega garantizada																					
TÉCNICAS UTILIZADAS																					
Igual que canal punto a punto confiable.																					
PROBLEMAS ENCONTRADOS O NO RESUELTOS																					
<p>Este caso funciona de manera correcta utilizando servicios web one-way. El problema viene al utilizar servicios web request-response. Lo que se hace es al crear la cola se le indica que se necesita respuesta, de esta manera:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p> Editar un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/PS2)</p> <div style="background-color: #f2f2f2; padding: 5px; margin-bottom: 5px;">Configuración de Transporte de JMS</div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Tipo de Destino</td> <td> <input checked="" type="radio"/> Cola <input type="radio"/> Tema </td> </tr> <tr> <td>Se Necesita Respuesta</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Patrón de Respuesta</td> <td> <input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID </td> </tr> <tr> <td>Tipo de Mensaje de Respuesta</td> <td> <input type="radio"/> Bytes <input checked="" type="radio"/> Texto </td> </tr> <tr> <td>Política de Distribución</td> <td>default ▾</td> </tr> <tr> <td>Codificación de Peticiones</td> <td>UTF-8</td> </tr> <tr> <td>Codificación de Respuestas</td> <td>UTF-8</td> </tr> <tr> <td>Timeout de Respuesta de Cliente</td> <td>300</td> </tr> <tr> <td>URI de Respuesta</td> <td>jms://_ _ _ _ _/weblogic.jms.XA</td> </tr> <tr> <td>Cuenta de Servicio de JMS</td> <td> <input type="text"/> <input type="button" value="Examinar..."/> </td> </tr> </table> <div style="background-color: #f2f2f2; padding: 5px; margin-bottom: 5px;">Configuración Avanzada</div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> << Anterior Siguiente >> Último >> Cancelar </div> </div> <p>La aplicación cliente llama al proxy, y, a pesar de los fallos del servicio, todas las llamadas tienen éxito y el servicio de negocio recibe todos los mensajes (las respuestas llegan a la cola de respuesta pero se consumen por lo que el problema viene cuando otro servicio necesita acceder a esos mensajes de respuesta ya que los necesita como dato de entrada).</p> <p>Ahora se necesita que entre el servicio proxy y el de negocio tengamos un canal de mensajes de entrega garantizada como en el caso anterior pero al parar el servidor y volverlo a iniciar se vuelve loco y esta todo el rato devolviendo el siguiente warning:</p> <pre><27-ene-2011 09H05' CET> <Warning> <WliSbTransports> <BEA-381542> <JmsAsyncResponseMDB.onMessage() ha recibido un mensaje con un identificador de mensaje desconocido bb4a176d0000012dc66feaaeffff804a; marcando la transacciÃ³n para rollback...></pre>		Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema	Se Necesita Respuesta	<input checked="" type="checkbox"/>	Patrón de Respuesta	<input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID	Tipo de Mensaje de Respuesta	<input type="radio"/> Bytes <input checked="" type="radio"/> Texto	Política de Distribución	default ▾	Codificación de Peticiones	UTF-8	Codificación de Respuestas	UTF-8	Timeout de Respuesta de Cliente	300	URI de Respuesta	jms://_ _ _ _ _/weblogic.jms.XA	Cuenta de Servicio de JMS	<input type="text"/> <input type="button" value="Examinar..."/>
Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema																				
Se Necesita Respuesta	<input checked="" type="checkbox"/>																				
Patrón de Respuesta	<input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID																				
Tipo de Mensaje de Respuesta	<input type="radio"/> Bytes <input checked="" type="radio"/> Texto																				
Política de Distribución	default ▾																				
Codificación de Peticiones	UTF-8																				
Codificación de Respuestas	UTF-8																				
Timeout de Respuesta de Cliente	300																				
URI de Respuesta	jms://_ _ _ _ _/weblogic.jms.XA																				
Cuenta de Servicio de JMS	<input type="text"/> <input type="button" value="Examinar..."/>																				

Se ha enviado el problema al soporte de Oracle y se está a la espera de su respuesta.

CONCLUSIONES

En este caso es necesario crear una transacción global para evitar la pérdida de los mensajes.

Si se produce un error, el mensaje no se pierde sino que se guarda en una cola JMS. Permite realizar reintentos por si el error se ha dado en un momento puntual.

Es similar al caso de uso canal punto a punto confiable, la única diferencia es que en este caso además de permitir la confiabilidad, la entrega es garantizada. Se han enviado un conjunto de mensajes, pasan a la cola de espera, se ha parado el servidor de aplicaciones (weblogic), se ha vuelto a reiniciar y los mensajes que quedaban pendientes se han continuado con su procesamiento.

-Tabla 28. Informe: Canal de entrega garantizada-

Canal de mensajes fallidos (<i>dead letter channel</i>)
TÉCNICAS UTILIZADAS
<p><u>Manejo de colas JMS desde Weblogic</u></p> <p>Desde la consola de weblogic se permite pausar / reanudar la producción de las colas, pausar / reanudar su consumo así como pausar / reanudar la inserción de mensajes. También permite eliminar mensajes de una cola así como moverlos, exportarlos o importarlos.</p>
PROBLEMAS ENCONTRADOS O NO RESUELTOS
No se han encontrado.
CONCLUSIONES
<p>En este caso es necesario crear una transacción global para evitar la pérdida de los mensajes.</p> <p>Si se produce un error, el mensaje no se pierde sino que se guarda en una cola JMS. OSB no permite reprocesar los mensajes de la cola de error pero esto se puede realizar directamente desde weblogic. Nos permite mover los mensajes de la cola de error a la cola de peticiones para volver a procesarlos una vez que se ha resuelto el problema del <i>Web Service</i>. Esto se debe hacer “a mano”.</p>

-Tabla 29. Informe: Canal de mensajes fallidos (*dead letter channel*)-

<p align="center">Invocación de WS con tipos de datos propietarios de .Net desde Java</p>
<p>TÉCNICAS UTILIZADAS</p>
<p><u>Tecnologías para la transformación de mensajes.</u></p> <p>OSB agrega una capa de transformación de mensajes que es útil para:</p> <ul style="list-style-type: none"> • Mapear distintos datos entre el origen y el destino. • Manipular el contenido del mensaje. <p>OSB proporciona apoyo a la transformación de mensajes usando tres tecnologías estándares: XSLT, XPath y XQuery.</p> <p>Se puede utilizar XSLT para transformar datos XML de un esquema a otro. Del mismo modo, se puede utilizar XQuery y XPath para representar esquemas de transformación en documentos XML a XML, XML a no-XML, y no-XML a XML.</p> <p>Además, OSB apoya el uso de Message Format Language(MFL) para transformaciones en formatos no XML, como valores separados por comas (ficheros CSV), libros de copia COBOL, documentos de Intercambio Electrónico de Datos (EDI, Electronic Data Interchange), y así sucesivamente.</p>
<p>PROBLEMAS ENCONTRADOS O NO RESUELTOS</p>
<p>No se han encontrado.</p>
<p>CONCLUSIONES</p>
<p>Se han realizado las siguientes pruebas en relación a la invocación de servicios .Net con tipo de datos DataSet:</p> <ol style="list-style-type: none"> 1. <u>Llamada directa desde Java a un WS .Net con DataSet</u> <p>Intentamos consumir el WS .Net directamente desde Java. Al realizar la llamada desde Java el objeto del DataSet se define como 'any' lo cual genera dificultades al llamarlo desde Java, o cualquier otra tecnología que no sea Microsoft.</p> <ol style="list-style-type: none"> 2. <u>Llamada directa desde .Net a un WS .Net con DataSet</u> <p>La invocación se realiza de manera correcta ya que el tipo de datos es compatible (los dos son .Net).</p> <ol style="list-style-type: none"> 3. <u>Llamada desde Java al OSB</u> <p>Se ha publicado el WS .Net en el bus. En cuanto al bus, no muestra ningún problema en cuanto a la publicación de este WS. Intentamos consumirlo desde un cliente java y</p>

nos surge el mismo problema que si no lo publicamos en el bus. En este caso el bus no nos soluciona nada.

4. Llamada desde .Net al OSB

Se ha publicado el WS .Net en el bus. En cuanto al bus, no muestra ningún problema en cuanto a la publicación de este WS. Intentamos consumirlo desde un cliente .Net y accedemos de manera correcta ya que se trata de la misma tecnología.

5. Llamada desde Java al OSB con transformaciones XQuery

Al consumir un WS .Net con tipo de datos DataSet desde Java, el cliente Java recibe el xml pero no sabe cómo tratarlo, es decir, es un objeto 'any'. Para poder manejar este xml existen dos posibles caminos:

- Desarrollar un 'parser' en Java para el manejo del xml.
- Utilizar el OSB: se ha desarrollado un WSDL con los mismos objetos que el DataSet. Lo que permite realizar el OSB es un mapeo, es decir, una transformación de los datos devueltos por el WS .Net al tipo de datos esperado por el WS Java. Para ello se han utilizado transformaciones XQuery.

Una de las ventajas es que permite crear un 'parser' dentro del OSB sin tener que desarrollar código.

Otra ventaja es el 'desacople', es decir, el bus actúa como intermediario de manera que si por ejemplo añadimos un nuevo dato en la transformación no afecta de ninguna manera al cliente de Java. En el caso en que desarrollamos el parser directamente en el cliente de Java sería necesario modificar el cliente.

Además permite adaptar contratos WSDL, es decir, podemos adaptar los datos devueltos por un servicio al contrato que necesitemos, como se ha hecho en este ejemplo.

6. Llamada desde .Net al OSB con transformaciones XQuery

Accedemos de manera correcta al *Web Service* publicado por el OSB con las transformaciones requeridas desde .Net.

-Tabla 30. Informe: Invocación de WS con tipos de datos propietarios de .Net desde Java-

REFERENCIAS Y BIBLIOGRAFÍA

Referencias y Bibliografía

SOA

<http://www.oracle.com/us/technologies/soa/soa-suite-066466.html>

www.accenture.com/SiteCollectionDocuments/Local_Spain/PDF/SOA.pdf

- Open Source SOA. Jeff Davis. Ed Manning 2009
- Understanding Enterprise SOA. Eric Pulier and Hugh Taylor. Ed Manning 2005

OSB

<http://www.oracle.com/technetwork/middleware/service-bus/overview/index.html>

<http://www.oracle.com/technetwork/middleware/service-bus/documentation/index.html>

- The Definitive Guide to SOA: *Oracle Service Bus*, Second Edition. Jeff Davies, David Schorow, Samrat Ray y David Rieber. Apress 2008.

RUP

- El proceso unificado de desarrollo de software Ivar Jacobson, Grady Booch, James Rumbaugh

ANEXOS

Anexo I: ¿Qué tiene de diferente OSB 11.1.1.3.0 con respecto a las versiones anteriores?

En este anexo se describen las nuevas características que diferencia a *Oracle Service Bus 11g* (11.1.1.3.0) con las versiones anteriores:

1. Instalación y despliegue

La instalación de *Oracle Service Bus* es similar a la de otros productos Oracle SOA. El instalador es el instalador unificado de Oracle y la instalación de *Oracle Service Bus* se produce por separado, después de instalar *Oracle WebLogic Server* 10.3.3 con un instalador diferente.

Después de la instalación en un entorno de producción, se pueden crear esquemas de bases de datos utilizando el RCU (*Repository Creation Utility*).

En esta versión se puede instalar *Oracle Service Bus* y *Oracle SOA Suite* en el mismo *Middleware Home* (MW_HOME), y *Oracle Service Bus* y *Oracle SOA Suite* pueden coexistir en el mismo dominio.

2. Servicio de almacenamiento en caché de resultados con coherencia

Oracle Service Bus ahora soporta el servicio de almacenamiento en caché de resultados para *Business Services* con coherencia. Coherencia en Oracle es una solución de datos en memoria que permite a las organizaciones predecir a escala la misión crítica de las aplicaciones, proporcionando un acceso rápido a los datos de uso frecuente.

El servicio de almacenamiento en caché de resultados permite el almacenamiento en caché de la respuesta de un *Business Service*. Esto puede mejorar drásticamente el rendimiento si la respuesta del *Business Service* es relativamente estática. *Oracle Service Bus* utiliza una caché única para todos los *Business Service*. Sólo resultados válidos / correctos de los *Business Service* se almacenan en caché.

3. Transacción en flujo de mensajes

Con *Oracle Service Bus*, puedes configurar un *Proxy Service* para iniciar y ejecutar el flujo de mensajes en el contexto de una transacción global JTA para el transporte de entrada no transaccional (como HTTP y JMS con conexiones no-XA). En las versiones anteriores, el flujo de mensajes del *Proxy Service* se ejecuta en el contexto de una transacción global JTA sólo cuando el transporte de entrada empieza / propaga una transacción global (como JMS / XA o SB con la transacción propagada).

4. Transporte nativo Java EJB

El transporte JEJB permite exponer un *Proxy Service* como un '*stateless session bean*' (SLSB) junto con argumentos POJO directamente en el flujo del mensaje. Los

argumentos POJO pueden ser manipulados utilizando '*Java Callaouts*'. Los argumentos POJO pueden ser directamente enviados como argumentos a SLSBs invocados por el transporte de salida JEJB.

5. JMS

En *Oracle Service Bus*, el transporte JMS se ha mejorado para recibir y enviar objetos Java a y desde las colas JMS / temas.

6. Funciones XPath personalizadas

Oracle Service Bus proporciona un marco para envolver utilidades de código Java en funciones XPath, que puede ser reutilizado en líneas de expresiones XQuery, transformaciones XQuery *Mapper* usadas en flujos de mensajes y Split-Joins. Las funciones XPath personalizadas están disponibles en las herramientas de diseño (consola e IDE).

7. Adaptadores JCA

Oracle Service Bus ofrece soporte para *Oracle SOA Suite 11g Release1 Patch Set 2* (11.1.1.3.0) adaptadores JCA.

8. Transporte SOA-Direct

Oracle Service Bus proporciona transporte SOA-DIRECT para usarlo con *Oracle SOA Suite 11g* y siguientes. El transporte SOA-DIRECT permite la propagación de contextos transaccionales y seguros de *Oracle Service Bus* a *Oracle SOA Suite*.

El transporte SOA-DIRECT permite la conectividad nativa entre componentes de servicios de *Oracle Service Bus* y *Oracle SOA Suite*. *Oracle SOA Suite* provee un marco "*direct binding*" que permite exponer componentes de servicios de Oracle SOA Suite en una aplicación compuesta, y el transporte SOA-DIRECT interactúa con esto servicios expuestos, permitiendo que esos componentes de servicios interactúen en la capa del bus de servicios y aprovechar las capacidades y características de *Oracle Service Bus*. *SOA Suite* puede invocar un *Proxy Service* de *Oracle Service Bus* con un transporte SB.

9. Política OWSM

Oracle Service Bus permite el uso de políticas de seguridad de Oracle *Web Services Manager* basados en SOAP Proxy/*Business Services*.

10. Otras mejoras

Otras mejoras del producto incluyen:

11.1. EJB 3.0

En *Oracle Service Bus*, el nuevo transporte JEJB introducido y el transporte EJB existente permiten la invocación de EJBs 3.0. Además, el transporte JEJB permite exponer un *Proxy Service* como un EJB 3.0 *stateless session bean*.

11.2. Soporta WSIL 1.1

Oracle Service Bus permite descubrir WSDL basado en *Proxy Service* usando Web Services Inspection Language (WSIL) 1.1. Esto permite a un navegador WSIL, como el que se presenta en Oracle JDeveloper, descubrir *Proxy Service* de *Oracle Service Bus* y descargar los artefactos de una manera compatible con los estándares.

11.3. Mejora de escalabilidad de logeo de alertas

Oracle Service Bus proporciona una mejor escalabilidad de logeo de alertas de manera distribuida. El logeo de alertas puede ser activado o desactivado. La consola de logeo de alertas se ha mejorado para escalar bien un gran número de alertas.

11.4. Mejoras en Split-Join

Las mejoras de Split-Join incluyen:

- **Soporta transacciones** – Split-Join se puede ejecutar en el contexto de una transacción global JTA.
- **Espera** – Split-Join permite una acción de espera (Wait action) para detener la ejecución de una ramificación (branch) de una duración determinada.

11.5. Mejoras UDDI

Oracle Service Bus permite cambiar el nombre de los servicios de negocio creados por la importación de UDDI. Se han realizado mejoras en la importación y establecimiento de suscripciones automáticas a los servicios UDDI conteniendo múltiples plantillas de enlace.

11.6. Mejoras en otros transportes

Mejoras en otros transportes incluyen:

- **MQ Dynamic and Alias Queues** – *Oracle Service Bus* proporciona soporte para la correlación de la respuesta en servicios de negocio MQ petición/respuesta basados en colas MQ (respuesta) dinámicas, que se crean de forma dinámica sobre la base de una cola modelo. También permite recibir mensajes de entrada con transporte MQ de MQ Alias Queues.
- **JMS Failover for Request/Response Pairs** – Para los patrones CorrelationID y MessageID, *Oracle Service Bus* permite una mayor disponibilidad que permite especificar una URI de respuesta (y cuenta de servicio) para cada URI de petición en servicios de negocio JMS de entrada/salida.
- **Email Load Balancing** – *Oracle Service Bus* permite especificar varios extremos URL (direcciones de correo) para balanceo de carga.

Anexo II: Documentos de Implementación⁶

CASOS DE USO DEL GAF

Caso de Uso 1: Publicar y proteger con WS-Security un WS sin protección

Hacer que el Bus de Integración publique un servicio (*Proxy Service* en nomenclatura OSB) protegido con *WS Security (UserNameToken)*. El servicio de backend (*Business Service* en nomenclatura OSB) sería un *Web Service* abierto, sin protección.

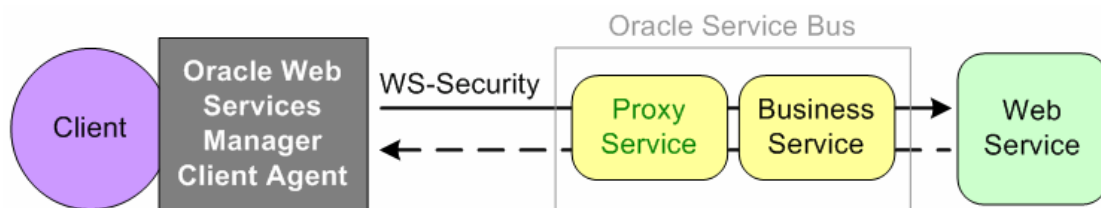
Los usuarios utilizados para autenticarse en la llamada al *Web Service* deberán ser de Directorio Activo (cuentas de servicio).

Elaborar un proyecto Web simple tanto en .NET como en Java que invoque el *Web Service* publicado por el OSB.

WSDL utilizado:

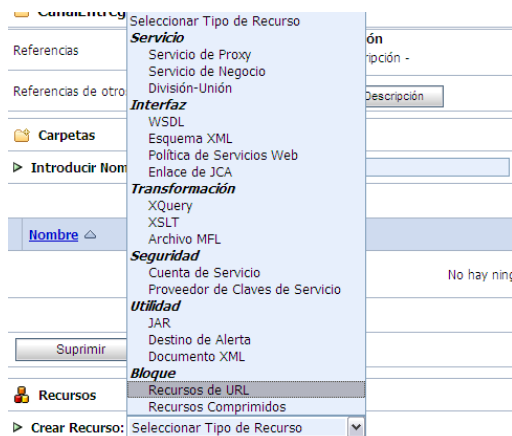
<http://a.b.c.d:8081/WS-WSDescrpcion-context-root/DescrpcionPort?WSDL>

Securización de un servicio a nivel de mensaje



Creamos un nuevo proyecto con el nombre WSSecurity. Creamos tres nuevas carpetas: *Business Services*, *Proxy Services* y Recursos.

Añadimos el WSDL a la carpeta recursos:



Añadimos la URL del recurso:

⁶ IPs y algunos datos como usuarios y contraseñas han sido ocultados por confidencialidad

Cargar Recursos

Cargar Recursos desde URL

URL/ Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

“Siguiete”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 1 de 1

<input checked="" type="checkbox"/>	Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	WS-WSDescription-context-root/WS-WSDescription-...	DescripcionPort	WSSecurity/Recursos	WSDL

Elementos 1 - 1 de 1

“Importar”.

Añadimos un nuevo servicio de negocio en la carpeta *Business Services*:

Seleccionar Tipo de Recurso

Servicio

- Servicio de Proxy
- Servicio de Negocio
- División-Unión

Interfaz

- WSDL
- Esquema XML
- Política de Servicios Web
- Enlace de JCA

Transformación

- XQuery
- XSLT
- Archivo MFL

Seguridad

- Cuenta de Servicio
- Proveedor de Claves de Servicio

Utilidad

- JAR
- Destino de Alerta
- Documento XML

Bloque

- Recursos de URL
- Recursos Comprimidos

Recursos

Crear Recurso:

Creamos el *Business Service* a partir del wsdl sin proteger:

Crear un Servicio de Negocio (WSecurity/Proxy Services/)

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL
 (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

“Último” y “Guardar”.

Generamos el *Proxy Service* a partir de este *Business Service* que hemos creado.

WSecurity/Proxy Services/

Referencias

Referencias de otros

Carpetas

Introducir Nombre

Nombre

Suprimir

Recursos

Crear Recurso:

Seleccionar Tipo de Recurso

- Servicio
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque
 - Recursos de URL
 - Recursos Comprimidos

Añadimos un nombre y lo creamos a partir del servicio de negocio que hemos creado.

Crear un Servicio de Proxy (WSecurity/Proxy Services/)

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL
 (puerto o enlace)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☒ Servicio de Negocio

☐ Servicio de Proxy

“Siguiente”. Introducimos la URI de punto final que queramos o dejamos la de por defecto.

“Último” y “Guardar”.

Seleccionamos la pestaña “Política” del proxy que acabamos de crear. Elegimos la siguiente política:

“Actualizar”.

Seleccionamos la pestaña “Seguridad”. Elegimos la opción “Procesar cabecera WS-Security = Sí”.

“Actualizar”.

Guardar todo y activar cambios.

Si miramos la información de la política que hemos elegido:

Esta política utiliza las credenciales de la cabecera de SOAP WS-Security UsernameToken para autenticar a los usuarios. Sólo se soporta el mecanismo de texto sin formato. Las credenciales se autentican en el almacén de identidades configurado. Esta política se puede asociar a cualquier punto final basado en SOAP.

Los credenciales los creamos desde la consola Enterprise Manager (FMW Control). Lo que se hace desde EM es crear un credencial de usuario y password a la que le das un nombre. Luego se referencia dicha credencial en la prueba de la consola del OSB con dicho nombre. De esa manera, en la consola de test NO necesitas especificar el usuario y password si no que el producto lo recupera de dicha credencial.

Para poder crear los credenciales el primer paso es crear el keystore donde se van a ubicar dichos credenciales. Para ello seguimos los siguientes pasos:

1. Nos situamos en la siguiente ruta: <domain_home>\config\fmwconfig. Esta es la localización por defecto de los ficheros de keystore.
2. Crear un nuevo certificado de clave privada y el auto-firmado. Utilizar el comando genKey para crear una clave privada. Se creará una nueva clave privada si no existe. El siguiente comando genera una clave RSA, con RSA-SHA1 como el algoritmo de firma, con la prueba de alias en el almacén de claves default-keystore.jks.

```
keytool -genkey -alias test -keyalg "RSA" -sigalg "SHA1withRSA" -dname "CN=test, C=US" -keystore default-keystore.jks
```

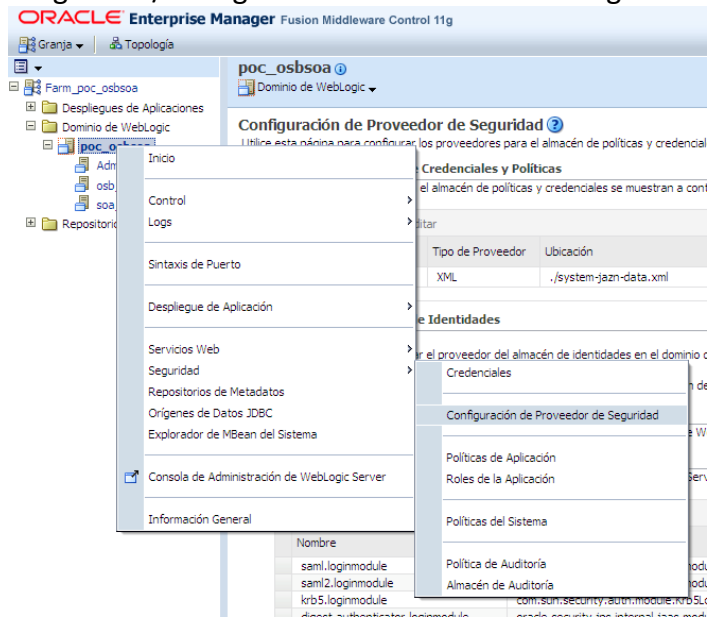
La utilidad keytool pide la clave necesaria y las contraseñas del almacén de claves. Clave DSA no está soportada. Asegúrate de pasar el parámetro "-keyalg RSA" en el comando.

3. Visualizar el almacén de claves. El siguiente comando muestra el contenido del almacén de claves. Se pedirá la contraseña del almacén de claves.

```
keytool -list -v -keystore default-keystore.jks
```

Una vez creado el keystore, configuramos la información del keystore desde la consola Enterprise Manager.

Dentro de la consola de EM (<http://a.b.c.d:7002/em>), abrir “Dominio de WebLogic” y hacer botón derecho sobre el nombre del dominio, para luego seleccionar “Seguridad/Configuración de Proveedor de Seguridad”:



Proveedor del Almacén de Identidades / Almacén de claves / Configurar

Configuración de Proveedor de Seguridad

Utilice esta página para configurar los proveedores para el almacén de políticas y credenciales del dominio de WebLogic, el almacén de claves y los módulos de conexión que utiliza Web Services Manager.

Proveedores de Almacén de Credenciales y Políticas

Los proveedores actuales para el almacén de políticas y credenciales se muestran a continuación. Para cambiar proveedores actuales para el almacén de políticas y credenciales, utilice el botón Cambiar Asociación.

Cambiar Asociación		Editar	
Nombre del Proveedor	Tipo de Proveedor	Ubicación	Almacén de Credenciales y Políticas
system-jazn-data.xml	XML	./system-jazn-data.xml	<input checked="" type="checkbox"/>

Proveedor del Almacén de Identidades

Para configurar y gestionar el proveedor del almacén de identidades en el dominio de WebLogic, utilice el [proveedor de seguridad de Oracle WebLogic Server](#).

Configure los parámetros para que la API de Rol y Usuario interactúe con el almacén de identidades. [Configurar...](#)

Proveedores de Autenticación de Web Services Manager

Puede configurar módulos de conexión y el almacén de claves para la autenticación de Web Services Manager.

Módulos de Conexión

La siguiente tabla enumera todos los módulos de conexión configurados para Web Services Manager. Utilice esta lista para crear, configurar o suprimir un módulo de conexión.

Crear...		Editar...		Suprimir...	
Nombre	Clase	Indicador de Control	Descripción		
saml.loginmodule	oracle.security.jps.internal.jaas.module.saml.JpsSAMLLoginMod...	Necesario	SAML Login Module		
saml2.loginmodule	oracle.security.jps.internal.jaas.module.saml.JpsSAML2LoginMo...	Necesario	SAML2 Login Module		
krb5.loginmodule	com.sun.security.auth.module.Krb5LoginModule	Necesario	Kerberos Login Module		
digest.authenticator.loginmodule	oracle.security.jps.internal.jaas.module.digest.DigestLoginMod...	Necesario	Digest Authenticator Login Module		
certificate.authenticator.loginmodule	oracle.security.jps.internal.jaas.module.x509.X509LoginModule	Necesario	X509 Certificate Login Module		
wss.digest.loginmodule	oracle.security.jps.internal.jaas.module.digest.WSSDigestLogin...	Necesario	WSS Digest Login Module		
user.authentication.loginmodule	oracle.security.jps.internal.jaas.module.authentication.JpsUser...	Necesario	User Authentication Login Module		
user.assertion.loginmodule	oracle.security.jps.internal.jaas.module.assertion.JpsUserAsse...	Necesario	User Assertion Login Module		

Almacén de Claves

Utilice esta sección para especificar el almacén de claves utilizado para almacenar claves públicas y privadas para todas las conexiones seguras en el dominio de WebLogic. [Configurar...](#)

Tipo JKS

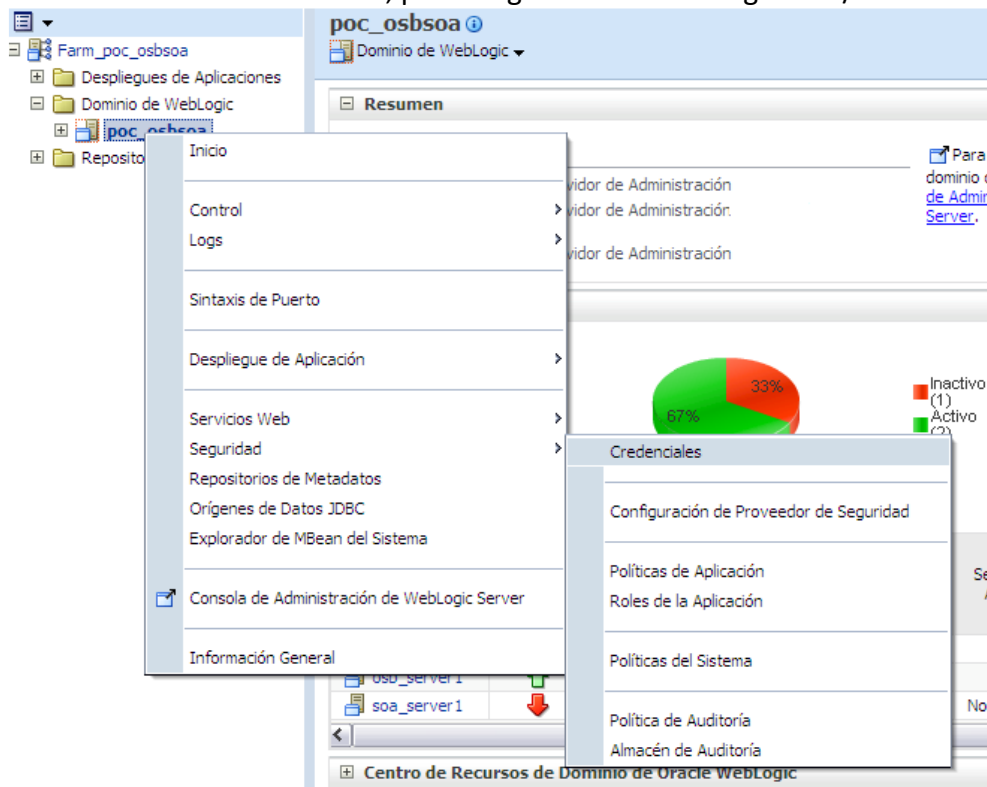
Ruta de Acceso /opt/oracle/middleware/domains/poc_osbsoa/config/fmwconfig/default-keystore.jks

Marcamos Configurar gestión de almacén de claves. Introducimos la ruta donde hemos creado el keystore. Introducimos una contraseña para el almacén de claves. Introducimos el Alias que en este caso le hemos llamado test y la contraseña que hemos puesto al crear el keystore.

Aceptar.

Ya se ha configurado el keystore. A continuación se detalla cómo crear los credenciales en dicho keystore:

Dentro de la consola de EM, abrir “Dominio de WebLogic” y hacer botón derecho sobre el nombre del dominio, para luego seleccionar “Seguridad/Credenciales”:



Después, dar al botón de “Crear Clave” y, asegurándote que la asignación seleccionada es la de “oracle.wsm.security”, crear una nueva “Contraseña” con el usuario y password: weblogic/<no mostrada por seguridad> (usuario y contraseña reales):

Editar Clave

Seleccionar Asignación: oracle.wsm.security
 Clave: prueba
 Tipo: Contraseña
 * Usuario: weblogic
 * Contraseña: ●●●●●●
 Descripción:

Aceptar Cancelar

Por último, la “Clave” (nombre) que se haya utilizado en el paso anterior es la que se debe proporcionar en el “Valor de Sustitución” de la consola de test del OSB:

Seguridad

Valores de Sustitución

Nombre de la Política	Propiedad	Valor por Defecto	Valor de Sustitución	Acciones
oracle/wss_username_token_client_policy	csf-key	basic.credentials	prueba	

Agregar

Prueba

1. Lanzamos el debugger del servicio “ProxyService1”.
2. En la ventana emergente, indicamos el valor de sustitución “prueba”.

Seguridad

Valores de Sustitución

Nombre de la Política	Propiedad	Valor por Defecto	Valor de Sustitución	Acciones
oracle/wss_username_token_client_policy	csf-key	basic.credentials	prueba	

Agregar

3. Nos devuelve:

Documento de Petición

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <wsd:GetDescriptionElement xmlns:wsd="http://wsdescription/">
      <wsd:description>string</wsd:description>
    </wsd:GetDescriptionElement>
  </soapenv:Body>
</soapenv:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:id="UsernameToken-hYUxAgU1qZYbK1UY6oGA22" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>weblogic</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">weblogic1</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soapenv:Body>
    <wsd:GetDescriptionElement xmlns:wsd="http://wsdescription/">
      <wsd:description>string</wsd:description>
    </wsd:GetDescriptionElement>
  </soapenv:Body>
</soapenv:Envelope>

```

Documento de Respuesta

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <m:GetDescriptionResponseElement xmlns:m="http://wsdescription/">
      <m:return>Descripcion pasada al servicio: string</m:return>
    </m:GetDescriptionResponseElement>
  </env:Body>
</env:Envelope>

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <m:GetDescriptionResponseElement xmlns:m="http://wsdescription/">
      <m:return>Descripcion pasada al servicio: string</m:return>
    </m:GetDescriptionResponseElement>
  </env:Body>
</env:Envelope>

```

4. Repetimos el test utilizando en este caso un valor de sustitución inexistente por ejemplo “pru”.

Seguridad

Valores de Sustitución

Nombre de la Política	Propiedad	Valor por Defecto	Valor de Sustitución	Acciones
oracle/wss_username_token_client_policy	csf-key	basic.credentials	pru	

Agregar

5. Nos devuelve:

Error al Llamar al Servicio de Prueba - ProxyService1

Ayuda

[OSB Security - OWSM:387256]Invalid CSF Key 'pru' set for override 'csf-key'. This CSF Key does not exist in the Credential Store.

Atrás

Cerrar

Si miramos el wsdl creado por el Proxy Service (http://a.b.c.d:8081/WSecurity/Proxy_Services/ProxyService1?wsdl), se puede ver la seguridad aplicada:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="DescriptionService" targetNamespace="http://wsdescription/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:WLSG3N2="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:WLSG3N0="http://schemas.xmlsoap.org/wsdl/" xmlns:WLSG3N1="http://wsdescription/"
- <wsp:Policy wsu:Id="wss_username_token_service_policy" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:oralgp="http://schemas.oracle.com/ws/2006/01/loggingpolicy" xmlns="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:orasp="http://schemas.oracle.com/ws/2006/01/securitypolicy"
  xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <sp:SupportingTokens xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
- <wsp:Policy>
- <sp:UsernameToken sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
- <wsp:Policy>
- <sp:WssUsernameToken10 />
- </wsp:Policy>
- </sp:UsernameToken>
- </wsp:Policy>
- </sp:SupportingTokens>
- </wsp:Policy>
- <wsdl:types>
- <xsd:schema elementFormDefault="qualified" targetNamespace="http://wsdescription/" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:WLSG3N3="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns="http://wsdescription/"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <xsd:element name="GetDescription" type="tns:GetDescription" />
- <xsd:complexType name="GetDescription">
- <xsd:sequence>
- <xsd:element name="description" nillable="true" type="string" />
- </xsd:sequence>
- </xsd:complexType>
- <xsd:element name="GetDescriptionResponse" type="tns:GetDescriptionResponse" />
- <xsd:complexType name="GetDescriptionResponse">
- <xsd:sequence>
- <xsd:element name="return" nillable="true" type="string" />
- </xsd:sequence>
- </xsd:complexType>
- <xsd:element name="GetDescriptionElement">
  <xsd:complexType>

```

Los usuarios utilizados para autenticarse en la llamada al *Web Service* deberán ser de Directorio Activo (cuentas de servicio).

Para ello necesitamos configurar el Security Provider ActiveDirectory. Entramos en la consola de weblogic.

Vamos a la estructura del dominio: Dominios de seguridad



Pulsar sobre el enlace del único reino de Seguridad existente “myrealm”.

Pestaña principal Proveedores. Se muestran los proveedores de seguridad por defecto. Añadimos uno nuevo. Pulsamos el botón “Nuevo”.

Creación de Nuevo Proveedor de Autenticación

Aceptar Cancelar

Crear Nuevo Proveedor de Autenticación

Las siguientes propiedades se utilizarán para identificar el nuevo proveedor de autenticación.

* Indica campos necesarios.

Nombre del proveedor de autenticación.

* **Nombre:**

Tipo de proveedor de autenticación que desea crear.

Tipo: ▼

Aceptar Cancelar

Nombre: AD-CuentasServicio

Type: ActiveDirectoryAuthenticator

Pulsar botón Aceptar.

Se muestra otra vez la lista de proveedores de seguridad.

Pulsamos el enlace del proveedor creado “ADAuthenticator”.

Pestaña Principal Configuración → Pestaña secundaria Común.

Valores para AD-CuentasServicio

Configuración Rendimiento

Común Proveedor Específico

Guardar

Esta página muestra información básica sobre este proveedor de autenticación de Active Directory. También puede utilizar esta página para definir el indicador de control de JAAS para controlar cómo se utiliza este proveedor en la secuencia de conexión.

Nombre:	AD-CuentasServicio	Nombre de este proveedor de autenticación de Active Directory. Más Información...
Descripción:	Provider that performs LDAP authentication	Breve descripción de este proveedor de autenticación de Active Directory. Más Información...
Versión:	1.0	Número de versión de este proveedor de autenticación de Active Directory. Más Información...
Indicador de Control:	<input type="text" value="SUFFICIENT"/> ▼	Especifica cómo se ajusta este proveedor de autenticación de Active Directory a la secuencia de conexión. Más Información...

Guardar

Indicador de control: SUFFICIENT

Pulsar botón Guardar.

Pestaña Principal Configuración → Pestaña secundaria Proveedor Específico.

Hay que configurar el proveedor de seguridad Active Directory con los valores adecuados al entorno.

Valores para AD-CuentasServicio

Configuración Rendimiento

Común Proveedor Específico

Guardar

Utilice esta página para definir la configuración específica de proveedor para este proveedor de autenticación de Active Directory.

Conexión

Host: Nombre de host o dirección IP del servidor LDAP. [Más Información...](#)

Puerto: Número de puerto en el que recibe el servidor LDAP. [Más Información...](#)

Principal: Nombre distinguido (DN) del usuario de LDAP que WebLogic Server debe utilizar para conectar con el servidor LDAP. [Más Información...](#)

Credencial: Credencial (normalmente una contraseña) utilizada para conectar con el servidor LDAP. [Más Información...](#)

Confirmar Credencial:

☐ **SSL Activado** Especifica si se debe utilizar el protocolo SSL al conectar al servidor LDAP. [Más Información...](#)

Usuarios

DN Base de Usuario: Nombre distinguido (DN) de la base del árbol en el directorio LDAP que contiene usuarios. [Más Información...](#)

Filtro de Todos los Usuarios: Si no se especifica el atributo (clase de objeto de usuario) (es decir, si el atributo es nulo o está vacío), se crea un filtro de búsqueda por defecto basado en el esquema de usuario. [Más Información...](#)

Filtro de Nombre de Usuario: Si no se especifica el atributo (atributo de nombre de usuario y clase de objeto de usuario) (es decir, si el atributo es nulo o está vacío), se crea un filtro de búsqueda por defecto basado en el esquema de usuario. [Más Información...](#)

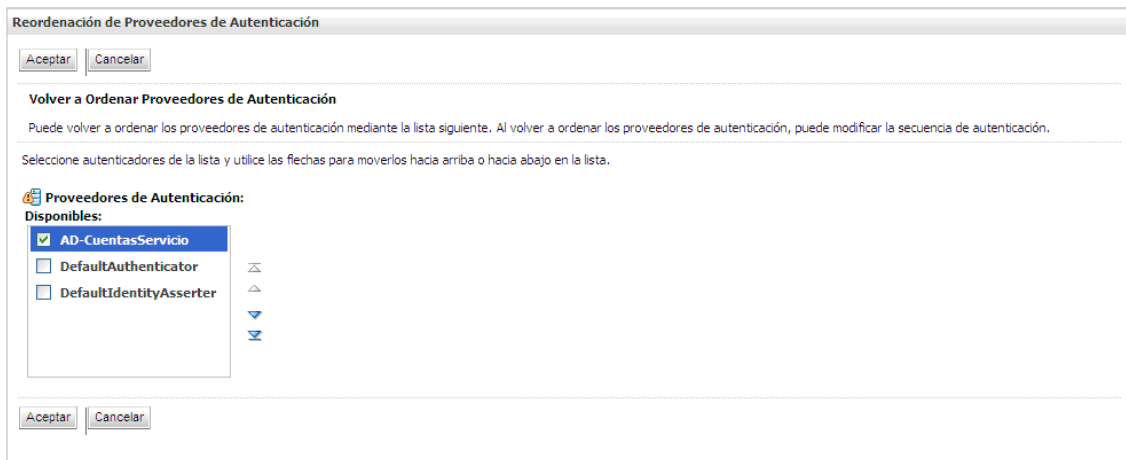
Ámbito de Búsqueda de Usuarios: Especifica el nivel de profundidad en el árbol de directorios LDAP en el que el proveedor de autenticación de LDAP debe buscar usuarios. [Más Información...](#)

Atributo de Nombre de Usuario: Atributo de un objeto de usuario LDAP que especifica el nombre del usuario. [Más Información...](#)

Clase de Objeto de Usuario: Clase de objeto LDAP que almacena usuarios. [Más Información...](#)

Pulsar botón Guardar.

Volver a la lista de Proveedores de seguridad. Reordenar los Proveedores de seguridad. Pulsar el botón Volver a ordenar.



Reordenarlo y poner el AD-CuentasServicio en primer lugar. Pulsar botón Aceptar. Reiniciar el servidor.

Prueba Cliente Java

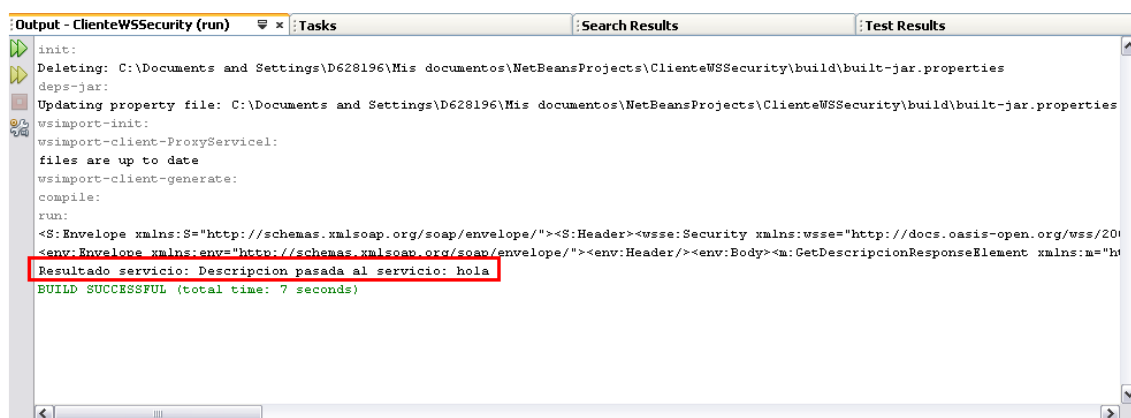
Cuenta de servicio de directorio activo utilizada

Usuario: NombreUsuario

Contraseña: <no mostrada por seguridad>

Se ha desarrollado un cliente java con Netbeans 6.9 (ClienteWSSecurity.zip). Lo que se ha hecho en el cliente es añadir a la cabecera SOAP el usuario y la contraseña para que sea reconocido por el *Web Service*.

Introducimos el usuario y contraseña anteriores. Resultado: tenemos acceso al servicio de negocio.



Si en la clase HeaderHandler.java cambiamos el usuario y contraseña a uno no existente, por ejemplo, usuario: NombreU, contraseña: <no mostrada por seguridad> (como el usuario anterior pero cambiando la contraseña).

Resultado: no tenemos acceso al servicio de negocio.

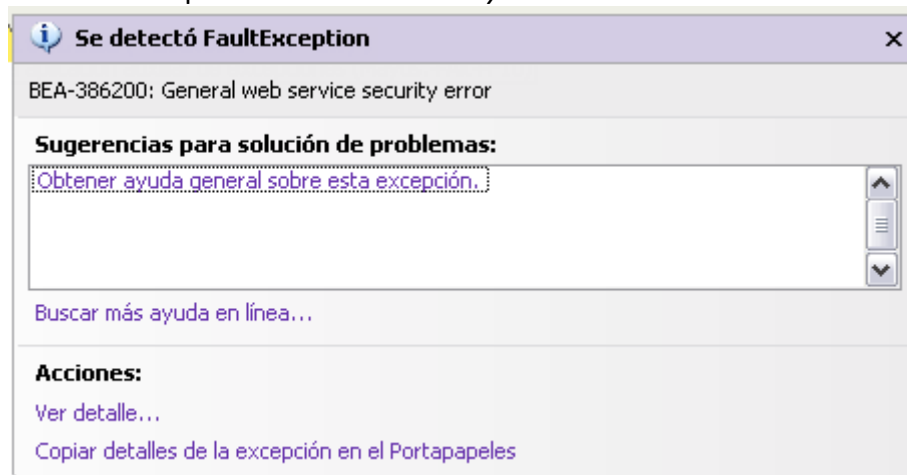
```

Output - ClienteWSSecurity (run)
wsimport-client-ProxyService1:
files are up to date
wsimport-client-generate:
Compiling 1 source file to C:\Documents and Settings\D628196\Mis documentos\NetBeansProjects\ClienteWSSecurity\build\classes
Note: C:\Documents and Settings\D628196\Mis documentos\NetBeansProjects\ClienteWSSecurity\src\clienteWSSecurity\HeaderHandler.java use:
Note: Recompile with -Xlint:unchecked for details.
compile:
run:
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Header><wssc:Security xmlns:wssc="http://docs.oasis-open.org/wss/2004/01/wss-schema" /></S:Header><S:Body></S:Body></S:Envelope>
Exception in thread "main" javax.xml.ws.soap.SOAPFaultException: BEA-386200: General web service security error
    at com.sun.xml.internal.ws.fault.SAPFault.getProtocolException(SAPFault.java:178)
    at com.sun.xml.internal.ws.fault.SAPFaultBuilder.createException(SAPFaultBuilder.java:119)
    at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:108)
    at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:78)
    at com.sun.xml.internal.ws.client.sei.SEIStub.invoke(SEIStub.java:107)
    at Proxy28.getDescription(Unknown Source)
    at clienteWSSecurity.Main.main(Main.java:25)
Java Result: 1
BUILD SUCCESSFUL (total time: 2 seconds)

```

Prueba Cliente .NET

La política que se ha utilizado en el caso 1 no es compatible con .Net ya que .Net 3.5 sólo es interoperable con WS-Security 1.1. El error es:



Para ello hemos añadido una política compatible con .Net:
wss1_username_token_with_message_protection_service_policy
Cambiamos la política anterior:



Actualizar.

Información de la política:

Esta política aplica protección a nivel de mensaje (integridad y confidencialidad de mensaje) y autenticación para solicitudes de SOAP entrantes de acuerdo con el estándar WS-Security v1.1. Los mensajes están protegidos mediante la serie Basic 128 de tecnologías de clave simétrica de WS-Security, en concreto los mecanismos de clave RSA para la confidencialidad del mensaje, el algoritmo hash SHA-1 para la integridad del mensaje y el cifrado AES de 128 bits. El almacén de claves se configura mediante la configuración de seguridad. Las credenciales se proporcionan mediante la cabecera de SOAP UsernameToken, de WS-Security Sólo se soporta el mecanismo de texto sin formato. Las credenciales se autentican en el almacén de identidades configurado. Esta política se puede asociar a cualquier punto final basado en SOAP.

Lanzamos el debugger del servicio “ProxyService1”.

Seguridad

Valores de Sustitución

Nombre de la Política	Propiedad	Valor por Defecto	Valor de Sustitución	Acciones
oracle/wss11_username_token_with_message...	keystore.recipient.alias	orakey	test	
	keystore.enc.csf.key	[Sin Políticas por Defecto]	prueba	
	csf-key	basic.credentials	prueba	

Agregar

Transporte

Anexo

Ejecutar Ejecutar-Guardar Restablecer Cerrar

En el primer valor de sustitución indicamos el alias que especifica los certificados de identidad por defecto (firma y claves de cifrado) para el keystore que hemos creado (default-keystore.jks), en este caso “test”. En los siguientes campos indicamos el nombre del credencial que hemos creado, en este caso “prueba”.

Creamos un cliente en .Net que utilice esta política (Prueba.zip).

Cuenta de servicio de directorio activo utilizada

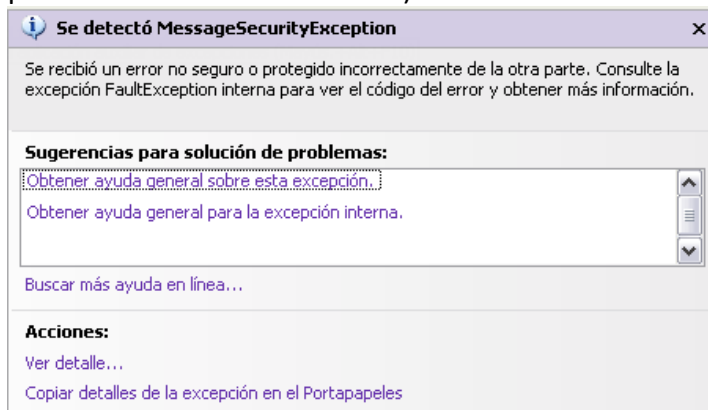
Usuario: NombreUsuario

Contraseña: <no mostrada por seguridad>

Resultado:



Si cambiamos el usuario y contraseña a uno no existente, por ejemplo, usuario: NombreUsuario, contraseña: <no mostrada por seguridad> (como el usuario anterior pero cambiando la contraseña). Resultado: no tenemos acceso al servicio de negocio.



Se puede realizar la misma prueba pero con transporte HTTPS, sólo hace falta cambiar en el proyecto del bus:

Configuración de Transporte de HTTP	
HTTPS Necesario	Sí
Autenticación	Ninguna

Explicación código cliente .Net tanto HTTP como HTTPS

Código de llamada al servicio

```
using System.Security.Cryptography.X509Certificates;
using System.Net;
using System.Net.Security;
using System.Configuration;

namespace Prueba
{
    public partial class Form1 : Form
    {
```

```

public Form1()
{
    InitializeComponent();
}

public bool ValidateServerCertificate(Object sender, X509Certificate certificate, X509Chain chain,
SslPolicyErrors sslPolicyErrors)
{
    return true;
}

private void button1_Click(object sender, EventArgs e)
{
    ServicePointManager.ServerCertificateValidationCallback = new
RemoteCertificateValidationCallback(ValidateServerCertificate);
    BUS.DescripcionClient cli = new BUS.DescripcionClient();
    cli.ClientCredentials.UserName.UserName = " xxxxx";
    cli.ClientCredentials.UserName.Password = xxxxx ";
    try
    {
        String resultado=cli.GetDescripcion("Hola").ToString();
        textBox1.Text = resultado;
        System.Console.WriteLine("FIN");
    }
    catch (Exception ex)
    {
        System.Console.WriteLine("ERROR:" + ex);
    }
}
}
}

```

A continuación se debe configurar el fichero app.config (si es una aplicación web lo haremos en el web.config):

- Debemos indicar el certificado que se va a utilizar (en este el certificado de “test”) y su localización. Ese certificado en formato .cer hay que guardarlo en la máquina que va a hacer la llamada para que pueda cifrar los mensajes, porque esta protección es por mensaje.

Para exportar el certificado X.509 del keystore en un archivo .cer:

```
keytool -export -alias test -file C:\test.cer -keystore default-keystore.jks
```

```

<behaviors>
  <endpointBehaviors>
    <behavior name="secureBehaviour">
      <clientCredentials>
        <serviceCertificate>
          <defaultCertificate findValue="4d 34 17 3a"
            storeLocation="LocalMachine" storeName="My" x509FindType="FindBySerialNumber"/>
        </serviceCertificate>
      </clientCredentials>
    </behavior>
  </endpointBehaviors>
</behaviors>

```

- Añadimos una nueva etiqueta “security” donde indicamos el tipo de autenticación que utiliza y la versión de seguridad del mensaje correspondiente a la política que utilizamos.

```

<customBinding>
  <binding name="ServiceBUS">
    <security defaultAlgorithmSuite="Basic128" authenticationMode="UserNameForCertificate"
      requireDerivedKeys="false" securityHeaderLayout="Lax" includeTimestamp="true"
      keyEntropyMode="CombinedEntropy" messageProtectionOrder="SignBeforeEncrypt"
      messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebruary2005WSecurityPolicy"
      requireSignatureConfirmation="true">
      <secureConversationBootstrap />
    </security>
    <textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16"
      messageVersion="Soap11" writeEncoding="utf-8">
      <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
        maxBytesPerRead="4096" maxNameTableCharCount="16384" />
    </textMessageEncoding>
    <httpsTransport manualAddressing="false" maxBufferPoolSize="524288"
      maxReceivedMessageSize="65536" allowCookies="false" authenticationScheme="Anonymous"
      bypassProxyOnLocal="false" hostNameComparisonMode="WeakWildcard"
      keepAliveEnabled="true" maxBufferSize="65536" proxyAuthenticationScheme="Anonymous"
      realm="" transferMode="Buffered" unsafeConnectionNtlmAuthentication="false"
      useDefaultWebProxy="true" />
    </binding>
  </customBinding>

```

En el caso de HTTP, debemos:

- Indicar que lo que tenemos es transporte HTTP:

```

<textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16"
  messageVersion="Soap11" writeEncoding="utf-8">
  <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
    maxBytesPerRead="4096" maxNameTableCharCount="16384" />
</textMessageEncoding>
<httpTransport manualAddressing="false" maxBufferPoolSize="524288"
  maxReceivedMessageSize="65536" allowCookies="false" authenticationScheme="Anonymous"
  bypassProxyOnLocal="false" hostNameComparisonMode="WeakWildcard"
  keepAliveEnabled="true" maxBufferSize="65536" proxyAuthenticationScheme="Anonymous"
  realm="" transferMode="Buffered" unsafeConnectionNtlmAuthentication="false"
  useDefaultWebProxy="true" />
</binding>

```

- Cambiar la URL:

```

<client>
  <endpoint address="http://localhost:8080/WSecurity/Proxy Services/ProxyService1?wsdl"
    behaviorConfiguration="secureBehaviour" binding="customBinding"
    bindingConfiguration="ServiceBUS" contract="BUS.Description"
    name="DescripcionPort1">
    <identity>
      <dns value="test" />
    </identity>
  </endpoint>
</client>

```


- Cambiar la referencia al servicio:

http://a.b.c.d:8081/WSecurity/Proxy_Services/ProxyService1?wsdl

En el caso de HTTPS, debemos:

- Indicar que lo que tenemos es transporte HTTPS:

```
<textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16"
  messageVersion="Soap11" writeEncoding="utf-8">
  <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
    maxBytesPerRead="4096" maxNameTableCharCount="16384" />
</textMessageEncoding>
<httpsTransport manualAddressing="false" maxBufferPoolSize="524288"
  maxReceivedMessageSize="65536" allowCookies="false" authenticationScheme="Anonymous"
  bypassProxyOnLocal="false" hostNameComparisonMode="WeakWildcard"
  keepAliveEnabled="true" maxBufferSize="65536" proxyAuthenticationScheme="Anonymous"
  realm="" transferMode="Buffered" unsafeConnectionNtlmAuthentication="false"
  useDefaultWebProxy="true" />
```

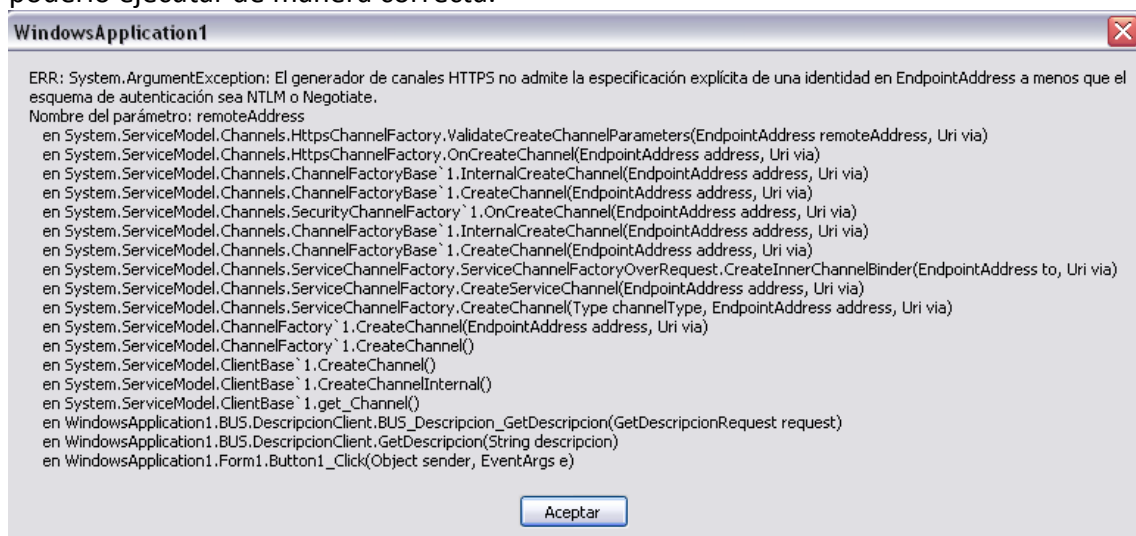
- Cambiar la URL:

```
<client>
  <endpoint address="https://a.b.c.d:8015/WSecurityNet/Proxy_Services/ProxyService1?wsdl"
    behaviorConfiguration="secureBehaviour" binding="customBinding"
    bindingConfiguration="ServiceBUS" contract="BUS.Description"
    name="DescriptionPort1">
    <identity>
      <dns value="test" />
    </identity>
  </endpoint>
</client>
```

- Cambiar la referencia al servicio:

https://a.b.c.d:8015/WSecurityNet/Proxy_Services/ProxyService1?wsdl

Al ejecutar la aplicación si sale el siguiente error, es necesario instalar .Net 3.5 SP1 para poderlo ejecutar de manera correcta.



Caso de Uso 2: Escalabilidad en servicios síncronos

Alcance:

El objetivo es demostrar la escalabilidad que proporciona contar con varios servicios de negocio iguales.

Paso 1

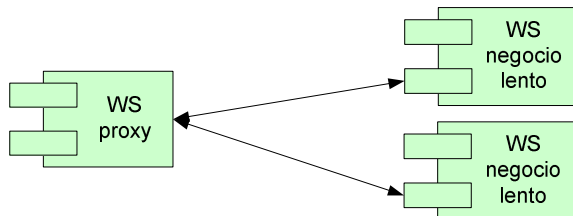
Tendremos un servicio Web SOAP de negocio que responderá de forma deliberadamente lenta.



Una aplicación cliente de prueba llamará al servicio proxy un número determinado de veces. Se tardará mucho tiempo en procesar todas las llamadas.

Paso 2

Ahora tendremos dos instancias del servicio de negocio lento.



La aplicación cliente llamará al proxy y se tardará, aproximadamente, la mitad de tiempo que antes en procesar todas las llamadas.

PASO 1

URL WSLENTO:

<http://a.b.c.d:8081/WSLento-WSlento-context-root/WSLENTOPort?WSDL>

Se ha desarrollado un *Web Service* que tarda 30 segundos en devolver una respuesta. De esta manera tenemos un servicio que siempre funciona pero de manera lenta.

Creamos un nuevo proyecto llamado EscalabilidadServiciosSincronosPaso1. Añadimos tres nuevas carpetas: *Business Services*, *Proxy Services* y Recursos.

En la carpeta Recursos añadimos el WSLENTO que hemos creado:

EscalabilidadServiciosSincronosPaso1/Recursos

Referencias

Referencias de otros recursos

Carpetas

Introducir Nombre

Nombre

Suprimir

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Seleccionar Tipo de Recurso

- Servicio**
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz**
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación**
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad**
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad**
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque**
 - Recursos de URL
 - Recursos Comprimidos

Introducimos la ruta del WSELETO:

Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

Siguiente >>

Cancelar

“Importar”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 2 de 2

Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
http://...	WSELETOPort	EscalabilidadServiciosSincronosPaso1/Rec...	WSDL
http://...	XMLSchema_1635946198	EscalabilidadServiciosSincronosPaso1/Rec...	Esquema XML

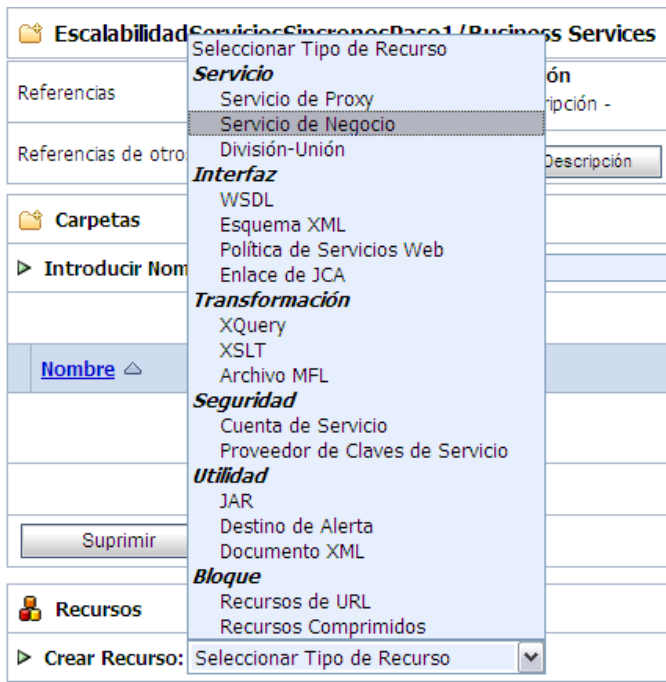
Elementos 1 - 2 de 2

<< Anterior

Importar

Cancelar

En la carpeta *Business Services* creamos el servicio de negocio que hace referencia al WSELETO que acabamos de añadir al proyecto.



Le damos un nombre (por ejemplo WSLento_BS) y lo creamos a partir de cualquier servicio SOAP:

Crear un Servicio de Negocio (EscalabilidadServiciosSincronosPaso1/Business Services/)

Configuración General

Nombre del Servicio* WSLento_BS

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☒ Servicio de Mensajes

☒ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> | Último >> | Cancelar

“Siguiente”.

Elegimos el Algoritmo de equilibrio de carga “Asignación en rueda (Round-Robin)”. Añadimos la URL del WSLENTO.

En el campo recuento de reintentos ponemos 1, es decir, si la URI a la que accedemos está offline (no disponible) lo vuelve a reintentar. En el caso de que hubiera dos URIs, reintentaría en la siguiente URI.

Editar un Servicio de Negocio (EscalabilidadServiciosSincronosPaso1/Business Services/WSLento_BS)

Configuración de Transporte

Protocolo*

Algoritmo de Equilibrio de Carga

URI de Punto Final*

URI EXISTENTES

Recuento de Reintentos

Intervalo de Iteración de Reintentos

Reintentar tras Errores de Aplicación ☒ Sí ☐ No

“Último” y “Guardar”.

Accedemos a la pantalla de configuración del servicio de negocio creado.

Recursos

▶ Crear Recurso:

Elementos 1 - 1 de 1

Nombre	Tipo de Recurso	Acciones	Opciones
WSLento_BS	Servicio de Negocio		

Elementos 1 - 1 de 1

Pestaña “Configuración operativa”.

Marcamos URI de punto final fuera de línea. Introducimos “20 segundos” por ejemplo, es decir, el intervalo de tiempo para el reintento.

Marcamos supervisión.

Marcamos Intervalo de agregación. Introducimos “3 minutos” por ejemplo, es decir, el tiempo en que se actualiza la información de supervisión del OSB.

Marcamos estado de regulación.

Simultaneidad máxima: Introducimos “9” por ejemplo, es decir, el número de mensajes tratados a la vez por cada URI.

Cola de regulación: Introducimos “1000” por ejemplo, es decir, el número máximo de mensajes en espera que se guardan en caso de que las URIs se encuentren ocupadas.

Detalles de Configuración **Configuración Operativa** Reglas de Alertas de SLA Políticas

Configuración General

Estado ☒ Activado

URI de Punto Final Fuera de Línea ☒ Activar con Intervalo de Reintentos 0 horas 0 mins. 30 segs.

Supervisión

Supervisión ☒ Activado

Intervalo de Agregación 0 horas 3 mins.

Alertas de SLA ☒ Activar Alertas en nivel Normal o superior

Regulación

Estado de Regulación ☒ Activado

Simultaneidad Máxima* 9

Cola de Regulación* 100 mensajes

Caducidad del Mensaje 0 milisegundos

Rastreo

Rastreo de Mensajes ☐ Activado

Nivel de Detalle Terse

Límite de Rastreo de Carga Útil 8 Kilobytes

Codificación por Defecto

Almacenamiento en Caché de Resultados

Estado de Caché de Resultados ☒ Activado

Atrás Actualizar Restablecer

“Actualizar”.

Lo que se ha hecho es configurar el servicio de negocio de manera que si un mensaje se envía a un extremo que no se encuentra disponible, el bus marca esta URI y lo intenta con la siguiente (según el número de reintentos configurados).

En la carpeta *Proxy Services* creamos el Proxy que llama al servicio de negocio que acabamos de crear:

Escalabilidad Servicios Sincronos Paso 1 / Proxy Services

Referencias

Referencias de otros

Carpeta

Introducir Nombre

Nombre

Suprimir

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Seleccionar Tipo de Recurso

Servicio

- Servicio de Proxy
- Servicio de Negocio
- División-Unión

Interfaz

- WSDL
- Esquema XML
- Política de Servicios Web
- Enlace de JCA

Transformación

- XQuery
- XSLT
- Archivo MFL

Seguridad

- Cuenta de Servicio
- Proveedor de Claves de Servicio

Utilidad

- JAR
- Destino de Alerta
- Documento XML

Bloque

- Recursos de URL
- Recursos Comprimidos

No hay ning

Ponemos un nombre (por ejemplo WSLento_PS) y creamos el Proxy a partir del WSLENTO creado:

Crear un Servicio de Proxy (EscalabilidadServiciosSincronosPaso2/Proxy Services/)

Configuración General

Nombre del Servicio* WSLento_PS

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL
 (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

“Siguiente”, “Último” y “Guardar”.

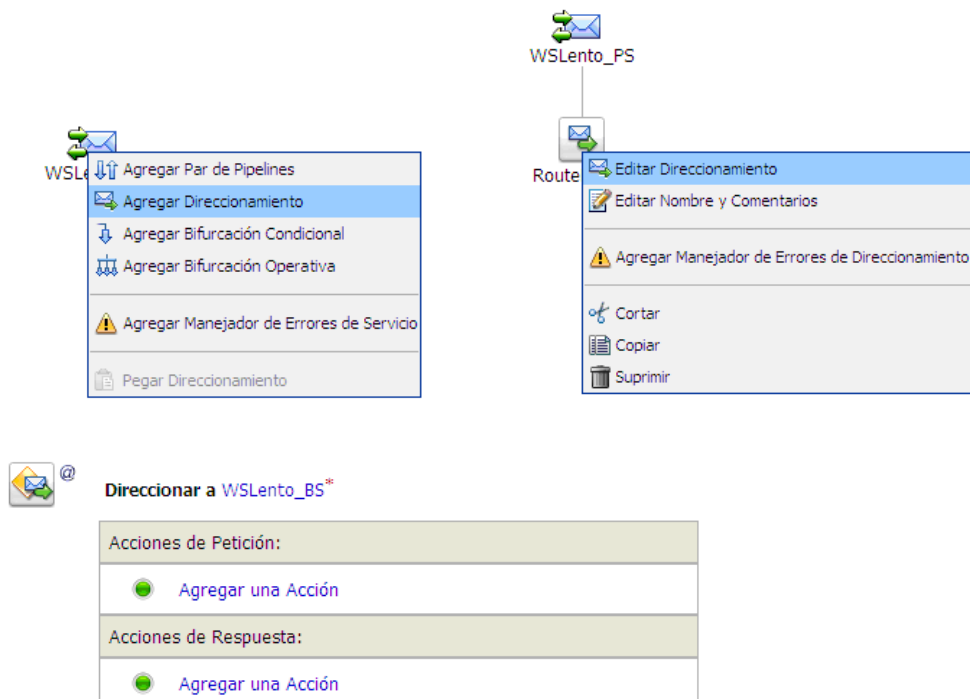
Flujo de mensajes.

Recursos

► **Crear Recurso:** Seleccionar Tipo de Recurso ▼

Nombre	Tipo de Recurso	Acciones	Opciones
WSLento_PS	Servicio de Proxy	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	añe <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Añadir “Route to” al servicio de negocio anterior:



Activar cambios.

Prueba

URL EscalabilidadServiciosSincronosPaso1 OSB:

http://a.b.c.d:8081/EscalabilidadServiciosSincronosPaso1/Proxy_Services/WSLento_PS?wsdl

No se pueden realizar las pruebas desde la consola del OSB ya que al ser servicios síncronos al hacer una llamada hay que esperar la respuesta para poder realizar otra llamada. Para ello se ha creado un cliente en java que nos permite realizar tres llamadas y nos muestra el momento en que se inicia cada llamada y el momento en que se finaliza.

Se puede probar desde:

<http://a.b.c.d:8081/CienteEscalabilidadSincronoPaso1/index.jsp>

Hello World!

Minuto inicio llamadas 18 Segundo inicio llamadas 36

El resultado se verá reflejado en el log del OSB. Por ejemplo, si ejecutamos una vez la aplicación obtenemos el siguiente resultado:

```
Minuto inicio 18
Segundo inicio 36
Minuto inicio 18
Segundo inicio 36
Minuto inicio 18
Segundo inicio 36
<12-ene-2011 14H19' CET> <Error> <Default> <BEA-000000> <Se ha detectado un estado de programa no válido
: Attempted to remove a WLSExecutionContext from the current WorkContextMap but the object in the Map was
not the object to be removed.>
Llamada a Servicio: OK, Servicio 1
Minuto fin 19
Segundo fin 6
<12-ene-2011 14H19' CET> <Error> <Default> <BEA-000000> <Se ha detectado un estado de programa no válido
: Attempted to remove a WLSExecutionContext from the current WorkContextMap but the object in the Map was
not the object to be removed.>
Llamada a Servicio: OK, Servicio 1
Minuto fin 19
Segundo fin 6
<12-ene-2011 14H19' CET> <Error> <Default> <BEA-000000> <Se ha detectado un estado de programa no válido
: Attempted to remove a WLSExecutionContext from the current WorkContextMap but the object in the Map was
not the object to be removed.>
Llamada a Servicio: OK, Servicio 1
Minuto fin 19
Segundo fin 6
```

Las tres llamadas se han tratado a la vez.

En la consola del OSB se puede llevar un seguimiento de las llamadas.

Accedemos a Operaciones / Estado de servicio.

Seleccionamos el servicio de negocio creado WSLento_BS.

ORACLE® Service Bus 11gR1

Bienvenido, weblogic: Conectado a: poc_osbsoa Inicio Consola de Oracle WLS

Alertas de SLA Alertas de Pipeline **Estado de Servicio** Estado del Servidor 1

Estado de Servicio

Mostrar Estadísticas Intervalo de Agregación Actual

Servidor osb_server1

Búsqueda

Nombre

Ruta de Acceso

Filtros de Estado de Servicio

Búsqueda Ver Todo

Nombre	Ruta de Acceso	Tipo de Servicio	Intervalo de Agregación	Tiempo Medio de Respuesta	Mensajes
Servicio de Seguimiento	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	0
Servicio de Seguimiento (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	0
Servicio Gestor de Soportes	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	0
Servicio Gestor de Soportes (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	0
WSLento_BS	EscalabilidadServiciosSincronosPaso1/Bus...	Servicio de Negocio	0 hora(s) 3 minuto(s)	30 segs. 10 msecs.	3
WSLento_BS	EscalabilidadServiciosSincronosPaso2/Bus...	Servicio de Negocio	0 hora(s) 3 minuto(s)	0 msecs.	0

Principio

Pestaña Métricas de servicio. Comprobamos que llegan 3 mensajes:

Métricas de Servicio Operaciones URI de Punto Final

General

Recuento de Alertas de SLA	Alertas de 0 ; Normal 10/01/11 11:03
Tiempo Mínimo de Respuesta	30 segs. 8 msecs.
Tiempo Máximo de Respuesta	30 segs. 12 msecs.
Tiempo Medio de Respuesta General	30 segs. 10 msecs.
Recuento de Mensajes	3
Recuento de Errores	0
Recuento de Failover	0
Ratio de Éxitos	100,00%
Ratio de Fallos	0,00%
Errores de WS-Security	0
Errores de Validación	N/A

Almacenamiento en Caché de Resultados

Recuento de Aciertos	0
Ratio de Aciertos	0,00%

Regulación

Tiempo Mínimo de Regulación	0 msecs.
Tiempo Máximo de Regulación	1 min. 29 msecs.
Tiempo Medio de Regulación	30 segs. 10 msecs.

Atrás Restablecer Estadísticas Refrescar

Pestaña URI de Punto Final. Comprobamos el reparto de los mensajes.

Métricas de Servicio Operaciones URI de Punto Final

Elementos 1 - 1

URI de Punto Final	Recuento de Mensajes	Recuento de Errores	Tiempo Mínimo de Respuesta	Tiempo Máximo de Respuesta	Tiempo Medio de Respuesta
http://10.238.7.6:8011/WSLento-WSlento-context-root/WSL...	3	0	30 segs. 8 msecs.	30 segs. 12 msecs.	30 segs. 10 msecs.

Elementos 1 - 1

Atrás Restablecer Estadísticas Refrescar

PASO 2

URL WSLENTO:

<http://a.b.c.d:8081/WSLento-WSLento-context-root/WSLENTOPort?WSDL>

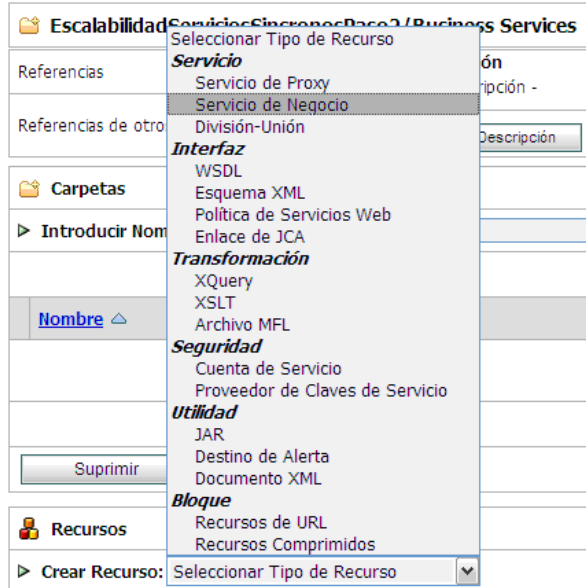
URL WSLENTO1:

<http://a.b.c.d:8081/WSLentoooo-WSLento-context-root/WSLENTOPort?WSDL>

Creamos un nuevo proyecto llamado EscalabilidadServiciosSincronosPaso2. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

Seguir pasos del PASO 1 hasta la creación de los recursos.

En la carpeta *Business Services* creamos un nuevo servicio de negocio.



Nombre del servicio “WSLento_BS” y se crea a partir de cualquier servicio SOAP.

Crear un Servicio de Negocio (EscalabilidadServiciosSincronosPaso2/Business Services/)

Configuración General

Nombre del Servicio*: WSLento_BS

Descripción:

Tipo de Servicio*:

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☒ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> Último >> Cancelar

“Siguiente”.

Elegimos el Algoritmo de equilibrio de carga “Asignación en rueda (Round-Robin)”.

Añadimos la URL del WSLENTO. Añadimos la URL del WSLENTO1.

En el campo recuento de reintentos ponemos 1, es decir, si la URI a la que accedemos está offline (no disponible) lo reintentamos con la siguiente.

Editar un Servicio de Negocio (EscalabilidadServiciosSincronosPaso2/Business Services/WSLento_BS)

Configuración de Transporte

Protocolo*:

Algoritmo de Equilibrio de Carga:

URI de Punto Final*

Formato:

URI EXISTENTES	OPCIONES
<input type="text" value="http://.../WSLento-WSLento-context-root/WSLENTOPort"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>
<input type="text" value="http://.../WSLentoooo-WSLento-context-root/WSLENTOPort"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>

Recuento de Reintentos:

Intervalo de Iteración de Reintentos:

Reintentar tras Errores de Aplicación: ☒ Sí ☐ No

“Último” y “Guardar”.

A partir de aquí seguir mismos pasos de PASO 1.

Prueba

URL EscalabilidadServiciosSincronosPaso2 OSB:

http://a.b.c.d:8081/EscalabilidadServiciosSincronosPaso2/Proxy_Services/WSLento_PS?wsdl

No se pueden realizar las pruebas desde la consola del OSB ya que al ser servicios síncronos al hacer una llamada espera la respuesta para poder realizar otra llamada. Para ello se ha creado un cliente en java que nos permite realizar tres llamadas y nos muestra el momento en que se inicia cada llamada y el momento en que se finaliza.

Se puede probar desde:

<http://a.b.c.d:8081/CienteEscalabilidadSincronoPaso2/index.jsp>

Escalabilidad en servicios síncronos

Minuto inicio llamadas 11 Segundo inicio llamadas 41

El resultado se verá reflejado en el log del OSB. Por ejemplo, si ejecutamos una vez la aplicación obtenemos el siguiente resultado:

Métricas de Servicio

URI de Punto Final

General

Recuento de Alertas de SLA	Alertas de 0 ; Normal 10/01/11 9:35
Tiempo Mínimo de Respuesta	30 segs. 9 msecs.
Tiempo Máximo de Respuesta	30 segs. 12 msecs.
Tiempo Medio de Respuesta General	30 segs. 10 msecs.
Recuento de Mensajes	3
Recuento de Errores	0
Recuento de Failover	0
Ratio de Éxitos	100,00%
Ratio de Fallos	0,00%
Errores de WS-Security	0
Errores de Validación	N/A

Almacenamiento en Caché de Resultados

Recuento de Aciertos	0
Ratio de Aciertos	0,00%

Regulación

Tiempo Mínimo de Regulación	0 msecs.
Tiempo Máximo de Regulación	29 segs. 973 msecs.
Tiempo Medio de Regulación	9 segs. 991 msecs.

Atrás

Restablecer Estadísticas



Refrescar

Pestaña URI de Punto Final. Comprobamos el reparto de los mensajes.

Métricas de Servicio

URI de Punto Final

Elementos 1 - 2

URI de Punto Final	Recuento de Mensajes	Recuento de Errores	Tiempo Mínimo de Respuesta	Tiempo Máximo de Respuesta	Tiempo Medio de Respuesta
 http://10.238.7.6:8011/WSLento-WSlento-context-root/WSL...	2	0	30 segs. 9 msecs.	30 segs. 9 msecs.	30 segs. 9 msecs.
 http://10.238.7.6:8011/WSLentoooo-WSlento-context-root/...	1	0	30 segs. 10 msecs.	30 segs. 10 msecs.	30 segs. 10 msecs.

Elementos 1 - 2

Atrás

Restablecer Estadísticas

Refrescar

Si paramos los dos *Web Services* desplegados en *Weblogic* y lanzamos la aplicación, obtenemos un mensaje de error.

Despliegues

Instalar

Actualizar

Suprimir

Iniciar

Parar

Mostrando 11 a 20 de 166 Anterior Siguiente

<input type="checkbox"/>	Nombre	Estado	Tipo	Orden de Despliegue
<input type="checkbox"/>	<div><div>+</div><div>AAWSBasadoContenido-WSServicio</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSDescripcion</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSDescripcionCompleta</div></div>	Activo	Aplicación Web	100
<input checked="" type="checkbox"/>	<div><div>+</div><div>AAWSLento-WSlento-context-root</div></div>	Activo	Aplicación Web	100
<input checked="" type="checkbox"/>	<div><div>+</div><div>AAWSLentoooo-WSlento-context-root</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSOrdenDePago</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSRegistro</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSRegistro2</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSRegistro4</div></div>	Activo	Aplicación Web	100
<input type="checkbox"/>	<div><div>+</div><div>AAWSsample1-Project1-context-root</div></div>	Activo	Aplicación Web	100

Instalar

Actualizar

Suprimir

Iniciar

Parar

Mostrando 11 a 20 de 166 Anterior Siguiente

```

Minuto inicio 29
Segundo inicio 35
Minuto inicio 29
Segundo inicio 35
Minuto inicio 29
Segundo inicio 35
<11-ene-2011 13H29' CET> <Error> <OSB Kernel> <BEA-380016> <Ninguno de los URI estÃ; en lÃ-nea para el servicio de negocio WSLento_BS>
Minuto inicio 29
Segundo inicio 36
Minuto inicio 29
Segundo inicio 36
Minuto inicio 29
Segundo inicio 36
<11-ene-2011 13H29' CET> <Error> <OSB Kernel> <BEA-380016> <Ninguno de los URI estÃ; en lÃ-nea para el servicio de negocio WSLento_BS>
<11-ene-2011 13H29' CET> <Error> <OSB Kernel> <BEA-380016> <Ninguno de los URI estÃ; en lÃ-nea para el servicio de negocio WSLento_BS>
<11-ene-2011 13H29' CET> <Error> <OSB Kernel> <BEA-380016> <Ninguno de los URI estÃ; en lÃ-nea para el servicio de negocio WSLento_BS>

```

Si volvemos a iniciar los *Web Services* no se continúa con la llamada sino que las peticiones se han perdido.

Si paramos un único *Web Services*, todas las peticiones serán tratadas por el *Web Services* que queda activo.

Caso de Uso 3: Escalabilidad en canal punto a punto

Alcance:

El objetivo es demostrar la escalabilidad que proporciona un canal punto a punto con varios consumidores iguales.

Paso 1

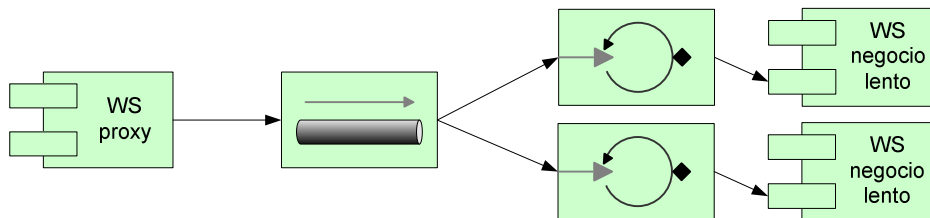
Tendremos un servicio Web SOAP de negocio de un solo sentido que consumirá los mensajes de forma deliberadamente lenta.



Una aplicación cliente de prueba llamará al servicio proxy un número determinado de veces. Se tardará mucho tiempo en procesar todas las llamadas.

Paso 2

Ahora tendremos dos instancias del servicio de negocio lento, y entre el proxy y los servicios un canal punto a punto.



La aplicación cliente llamará al proxy. Los mensajes se entregarán casi inmediatamente, y se tardará, aproximadamente, la mitad de tiempo que antes en que todos los mensajes sean consumidos.

PASO 1

URL WSLENTO:

<http://a.b.c.d:8081/WSLento-WSlento-context-root/WSLENTOPort?WSDL>

Creamos un nuevo proyecto llamado EscalabilidadCanalPuntoAPuntoPaso1. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

En la carpeta Recursos añadimos el WSLENTO que hemos creado:

Escalabilidad **Canal Punto A Punto / Business Services**

Referencias

Referencias de otro

Carpetas

► Introducir Nombre

Nombre

Suprimir

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Seleccionar Tipo de Recurso

- Servicio**
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz**
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación**
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad**
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad**
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque**
 - Recursos de URL
 - Recursos Comprimidos

Descripción

No

Introducimos la ruta del WSLENTO:

Cargar Recursos

Cargar Recursos desde URL

URL/ Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

Siguiente >> **Cancelar**

“Importar”.

Cargar Recursos

Revisar Recursos Cargados

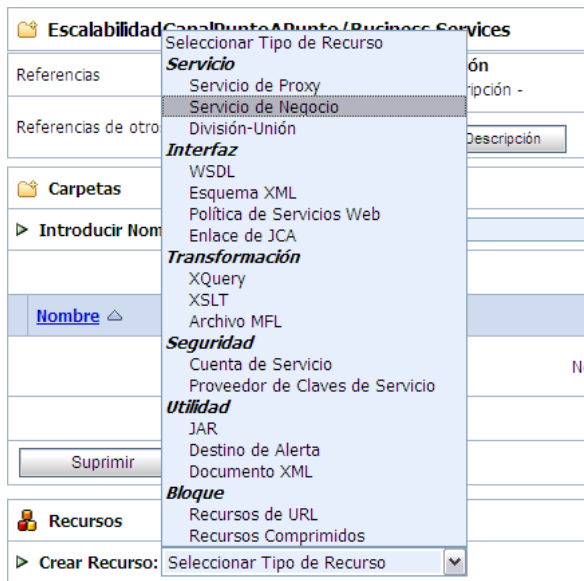
Elementos 1 - 2 de 2

<input checked="" type="checkbox"/>	Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	http://localhost:8080/WSLento-WSlento-C...	WSLENTOPort	EscalabilidadServiciosSincronosPaso1/Rec...	WSDL
<input checked="" type="checkbox"/>	http://localhost:8080/WSLento-WSlento-C...	XMLSchema_1635946198	EscalabilidadServiciosSincronosPaso1/Rec...	Esquema XML

Elementos 1 - 2 de 2

<< Anterior **Importar** **Cancelar**

En la carpeta *Business Services* creamos un nuevo servicio de negocio.



Nombre del servicio “WSLentoPuntoAPunto_BS” y se crea a partir de cualquier servicio SOAP.

Crear un Servicio de Negocio (EscalabilidadServiciosPuntoAPuntoPaso1/Business Services/)

Configuración General

Nombre del Servicio* WSLentoPuntoAPunto_BS

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☒ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> Último >> Cancelar

“Siguiente”.

Elegimos el Algoritmo de equilibrio de carga “Asignación en rueda”. Añadimos la URL del WSLENTO.

Crear un Servicio de Negocio (EscalabilidadServiciosPuntoAPuntoPaso1/Business Services/WSLentoPuntoAPunto_BS)

Configuración de Transporte

Protocolo* http

Algoritmo de Equilibrio de Carga asignación en rueda (round-robin)

URI de Punto Final*

Formato: http://host:port/someService

http://localhost:7001/EscalabilidadServiciosPuntoAPuntoPaso1/Business_Serv

Agregar

URI EXISTENTES

http://localhost:7001/WSLento-WSLento-context-root/WSLENTOPort

OPCIONES

Recuento de Reintentos 1

Intervalo de Iteración de Reintentos 30

Reintentar tras Errores de Aplicación ☒ Sí ☐ No

<< Anterior Siguiente >> Último >> Cancelar

“Último” y “Guardar”.

Accedemos a la pantalla de configuración del servicio de negocio creado.

Recursos

> Crear Recurso: Seleccionar Tipo de Recurso

Elementos 1 - 1 de 1

Nombre	Tipo de Recurso	Acciones	Opciones
WSLentoPuntoAPunto_BS	Servicio de Negocio		

Elementos 1 - 1 de 1

Suprimir

Pestaña “Configuración operativa”.

Marcamos URI de punto final fuera de línea. Introducimos “10 segundos” por ejemplo, es decir, el intervalo de tiempo para el reintento.

Marcamos supervisión.

Marcamos Intervalo de agregación. Introducimos “1 minuto” por ejemplo, es decir, el tiempo en que se actualiza la información de supervisión del OSB.

Marcamos estado de regulación.

Simultaneidad máxima: Introducimos “9” por ejemplo, es decir, el número de mensajes tratados a la vez por cada URI.

Cola de regulación: Introducimos “1000” por ejemplo, es decir, el número de mensajes que se guardan en caso de que las URIs se encuentren ocupadas.

Detalles de Configuración **Configuración Operativa** Reglas de Alertas de SLA Políticas

Configuración General

Estado ☒ Activado

URI de Punto Final Fuera de Línea ☒ Activar con Intervalo de Reintentos horas mins. segs.

Supervisión

Supervisión ☒ Activado

Intervalo de Agregación horas mins.

Alertas de SLA ☒ Activar Alertas en nivel Normal o superior

Regulación

Estado de Regulación ☒ Activado

Simultaneidad Máxima

Cola de Regulación mensajes

Caducidad del Mensaje milisegundos

Rastreo

Rastreo de Mensajes ☐ Activado

Nivel de Detalle Terse

Límite de Rastreo de Carga Útil Kilobytes

Codificación por Defecto

Almacenamiento en Caché de Resultados

Estado de Caché de Resultados ☒ Activado

“Actualizar”.

Lo que se ha hecho es configurar el servicio de negocio de manera que si un mensaje se envía a un extremo que no se encuentra disponible, el bus marca esta URI y lo intenta con la siguiente (según el número de reintentos configurados).

“Último” y “Guardar”.

A continuación creamos el Proxy WSLentoPuntoAPunto_PS2 a partir del WSLentoPuntoAPunto_BS.

Le damos un nombre: WSLentoPuntoAPunto_PS2 y lo creamos a partir de cualquier servicio SOAP. “Siguiente”.

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola que va a utilizar. Las colas se crean de manera automática una vez que ejecutemos el Proxy.

“Siguiente”.

Seleccionamos la siguiente configuración de transporte:

Editar un Servicio de Proxy (EscalabilidadCanalPuntoAPunto/Proxy Services/WSLentoPuntoAPunto_PS2)	
Configuración de Transporte de JMS	
Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema
Se Necesita Respuesta	<input checked="" type="checkbox"/>
Patrón de Respuesta	<input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID
Tipo de Mensaje de Respuesta	<input type="radio"/> Bytes <input checked="" type="radio"/> Texto
Política de Distribución	default
Codificación de Peticiones	UTF-8
Codificación de Respuestas	UTF-8
Timeout de Respuesta de Cliente	300
URI de Respuesta	ry/WSLentoPuntoAPunto_PS2Response
Cuenta de Servicio de JMS	<input type="text"/> Examinar...
Configuración Avanzada	
<input type="button" value="Anterior"/> <input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

Desplegamos Configuración avanzada.

Introducimos el número de reintentos en caso de fallo, el tiempo entre cada uno de esos reintentos (en milisegundos), el destino de error (en este caso la cola ColaError) y en caso de que después de todos los reintentos siga fallando se redirigiera a esta cola.

Configuración Avanzada	
Usar SSL	<input type="checkbox"/>
Selector de Mensajes	<input type="text"/>
Suscripción Duradera	<input type="checkbox"/>
Recuento de Reintentos	4
Intervalo de Reintentos	5000
Destino de Error	WSLentoPuntoAPunto_PS2Err
Política de Caducidad	Redireccionar
Se necesita XA	<input type="checkbox"/>
Timeout de JNDI	0
<input type="button" value="Anterior"/> <input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

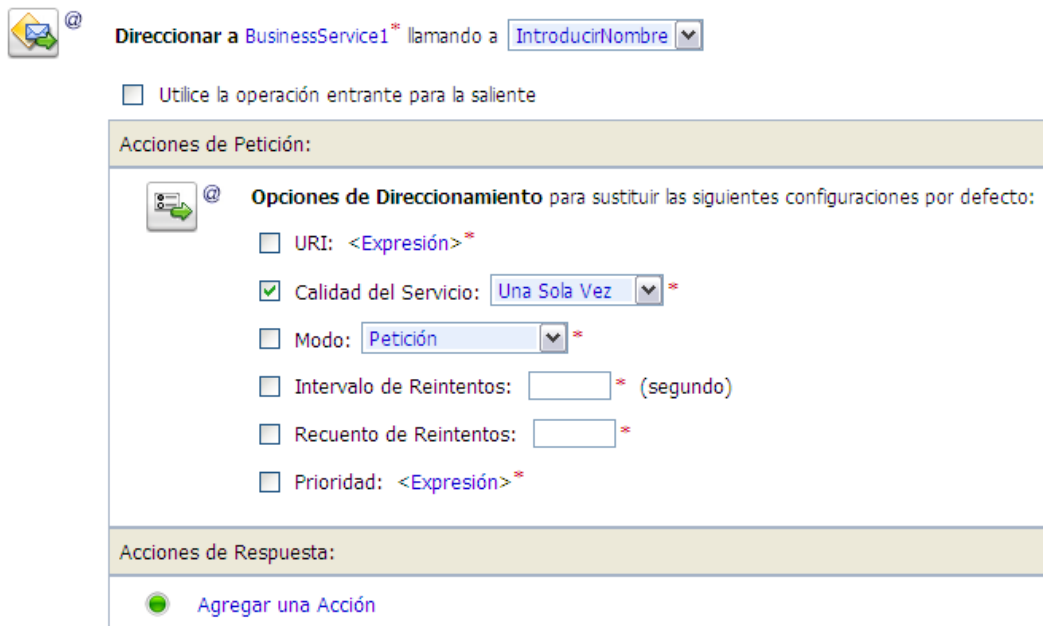
“Siguiente”.

“Siguiente”. Seleccionamos misma transacción para respuesta. Si algo ocurre se produce rollback en la transacción.

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)	
Manejo de Mensajes	
Transacción Necesaria	<input type="checkbox"/> Activado
Misma Transacción para Respuesta	<input checked="" type="checkbox"/> Activado
Flujo de Contenido	<input type="checkbox"/> Activado Tipo de Buffer <input type="radio"/> Buffer de Memoria <input type="radio"/> Buffer de Disco Compresión <input type="checkbox"/> Activado
<input type="button" value="Anterior"/> <input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

“Último” y “Guardar”.

Entramos en el flujo de mensajes del WSLentoPuntoAPunto_PS2 que acabamos de crear. “Editar direccionamiento” y añadimos “Opciones de direccionamiento”. Seleccionamos Calidad del Servicio “Una Sola Vez”, para evitar que se dupliquen los mensajes.



Direccionar a BusinessService1* llamando a **IntroducirNombre**

☐ Utilice la operación entrante para la saliente

Acciones de Petición:

Opciones de Direccionamiento para sustituir las siguientes configuraciones por defecto:

☐ URI: <Expresión>*

☒ Calidad del Servicio: **Una Sola Vez***


☐ Modo: **Petición***

☐ Intervalo de Reintentos: * (segundo)

☐ Recuento de Reintentos: *

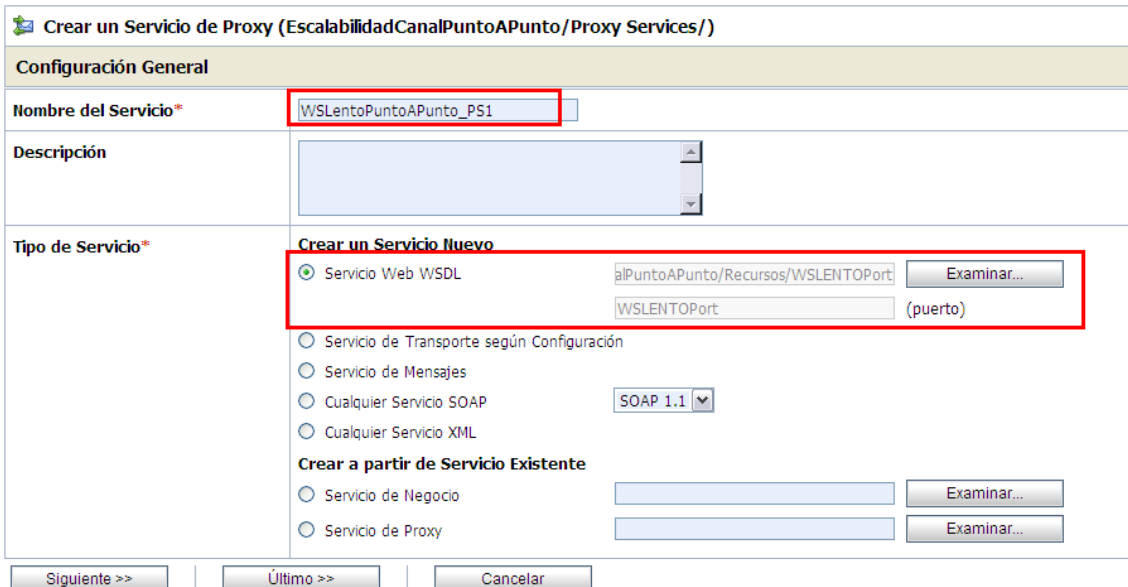
☐ Prioridad: <Expresión>*

Acciones de Respuesta:

 **Agregar una Acción**

A continuación creamos el Proxy principal al que se accede por HTTP y hace referencia al Proxy anteriormente creado.

Le damos un nombre (WSLentoPuntoAPunto_PS1) y lo creamos a partir del WSDL WSEntoPort:



Crear un Servicio de Proxy (EscalabilidadCanalPuntoAPunto/Proxy Services/)

Configuración General

Nombre del Servicio* **WSLentoPuntoAPunto_PS1**

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL **alPuntoAPunto/Recursos/WSEntoPort** **Examinar...**

WSEntoPort (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP **SOAP 1.1**

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio **Examinar...**

☐ Servicio de Proxy **Examinar...**

Siguiente >> **Último >>** **Cancelar**

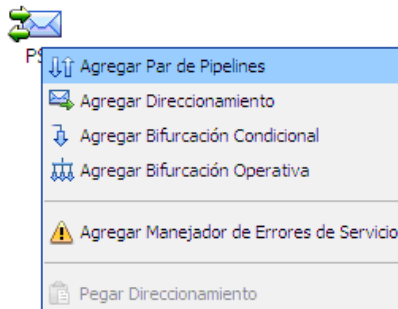
“Último” y “Guardar”.

Por último vamos al flujo de mensajes del WSLentoPuntoAPunto_PS1.

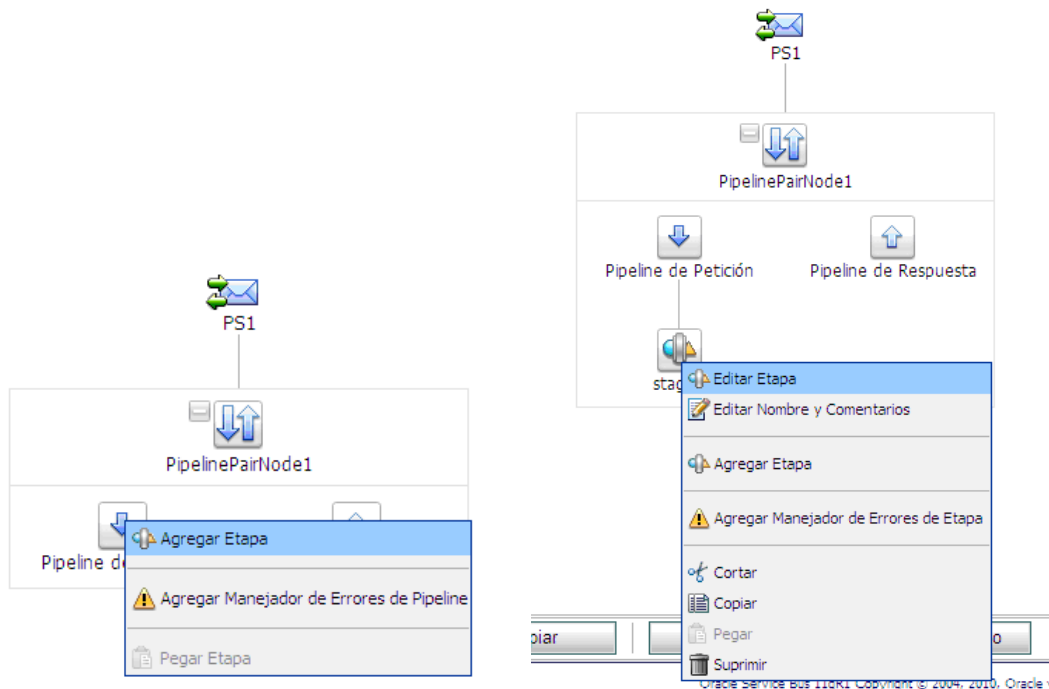
Recursos			
Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 2 de 2			
Nombre	Tipo de Recurso	Acciones	Opciones
WSLentoPuntoAPunto_PS1	Servicio de Proxy		aTe
WSLentoPuntoAPunto_PS2	Servicio de Proxy		aTe
Elementos 1 - 2 de 2			
Suprimir			

Añadimos un pipeline. Como queremos mandar el mensaje pero no queremos esperar a la respuesta sino obtener una respuesta inmediata debemos utilizar “Publish” en vez de “Route to”. Por ello:

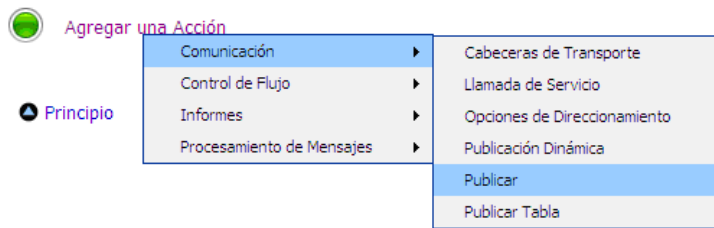
1. Añadimos un par de pipelines



2. En el pipeline de petición añadimos una nueva etapa.



3. Añadimos una nueva acción “Publicar”. Seleccionamos al servicio al que queremos direccionar, en este caso WSLentoPuntoAPunto_PS2.



Publicar en WSLentoPuntoAPunto_PS2**

Acciones de Petición:

 **Agregar una Acción**

Guardar todo y activar sesión.

Prueba

URL EscalabilidadServiciosSincronosPaso1 OSB:

http://a.b.c.d:8081/EscalabilidadServiciosPuntoAPuntoPaso1/Proxy_Services/WSLentoPuntoAPunto_PS1?wsdl

La aplicación se puede probar desde:

<http://a.b.c.d:8081/CienteEscalabilidadPuntoAPuntoPaso1/index.jsp>

Si ejecutamos la aplicación obtenemos el instante de la llamada por pantalla:

Escalabilidad en canal punto a punto

Minuto inicio llamadas 57 Segundo inicio llamadas 59

En la consola del OSB se puede llevar un seguimiento de las llamadas.

Accedemos a Operaciones / Estado de servicio.

Seleccionamos el servicio de negocio creado WSLentoPuntoAPunto_BS.

ORACLE® Service Bus 11gR1

Bienvenido, weblogic. Conectado a: poc_osboa Inicio | Consola de Oracle V

Alertas de SLA | Alertas de Pipeline | **Estado de Servicio** | Estado del Servidor 1

Estado de Servicio

Mostrar Estadísticas Intervalo de Agregación Actual

Servidor osb_server1

Búsqueda

Nombre

Ruta de Acceso

Filtros de Estado de Servicio

Búsqueda Ver Todo

Nombre	Ruta de Acceso	Tipo de Servicio	Intervalo de Agregación	Tiempo Medio de Respuesta	Me
Servicio de Seguimiento	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio de Seguimiento (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_BS	EscalabilidadCanalPuntoAPunto/Business S...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_BS	EscalabilidadServiciosPuntoAPuntoPaso1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_PS2	EscalabilidadCanalPuntoAPunto/Proxy Serv...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_PS2	EscalabilidadServiciosPuntoAPuntoPaso1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	

Principio

Pestaña Métricas de servicio. Comprobamos que llegan 3 mensajes:

Métricas de Servicio URI de Punto Final

General

Recuento de Alertas de SLA	Alertas de 0 ; Normal 12/01/11 9:59
Tiempo Mínimo de Respuesta	30 segs. 8 msecs.
Tiempo Máximo de Respuesta	30 segs. 11 msecs.
Tiempo Medio de Respuesta General	30 segs. 9 msecs.
Recuento de Mensajes	3
Recuento de Errores	0
Recuento de Failover	0
Ratio de Éxitos	100,00%
Ratio de Fallos	0,00%
Errores de WS-Security	0
Errores de Validación	N/A

Almacenamiento en Caché de Resultados

Recuento de Aciertos	0
Ratio de Aciertos	0,00%

Regulación

Tiempo Mínimo de Regulación	0 msecs.
Tiempo Máximo de Regulación	0 msecs.
Tiempo Medio de Regulación	0 msecs.

Atrás Restablecer Estadísticas Refrescar

Pestaña URI de Punto Final. Comprobamos el reparto de los mensajes.

Métricas de Servicio URI de Punto Final

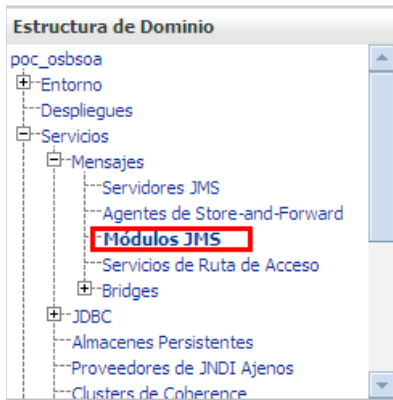
Elementos 1 - 1

URI de Punto Final	Recuento de Mensajes	Recuento de Errores	Tiempo Mínimo de Respuesta	Tiempo Máximo de Respuesta	Tiempo Medio de Resp...
http://10.238.7.6:8011/WSLento-WSlento-context-root/WSL...	3	0	30 segs. 8 msecs.	30 segs. 11 msecs.	30 segs. 9 msecs.

Elementos 1 - 1

Atrás Restablecer Estadísticas Refrescar

Para ver el momento en que llegan los mensajes vamos a Weblogic, Servicios / Módulos JMS.



Módulo JMS jmsResources.

Módulos JMS

Los recursos del sistema de JMS se configuran y almacenan como módulos parecidos a los módulos de J2EE estándar. Dichos recursos incluyen colas, temas, fábricas de conexiones, plantillas, claves de destino, cuotas, colas distribuidas, temas distribuidos, servidores ajenos y parámetros de Store-and-Forward (SAF) de JMS. Puede configurar y gestionar de forma administrativa módulos de sistema de JMS como recursos del sistema globales.

En esta página se resumen los módulos del sistema de JMS creados para este dominio.

[Personalizar esta Tabla](#)

Módulos JMS

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Tipo
<input type="checkbox"/>	BPMJMSModule	Sistema
<input type="checkbox"/>	jmsResources	Sistema
<input type="checkbox"/>	SOAJMSModule	Sistema
<input type="checkbox"/>	SystemModule	Sistema
<input type="checkbox"/>	UMSJMSSystemResource	Sistema
<input type="checkbox"/>	WseeJmsModule	Sistema

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

Seleccionamos WSLentoPuntoAPunto_PS2Respuesta.

Valores para jmsResources

Configuración Despliegues Secundarios Destinos Seguridad Notas

Esta página muestra información general sobre un módulo de sistema de JMS y sus recursos. También permite configurar nuevos recursos y acceder a los recursos existentes.

Nombre: jmsResources Nombre de este módulo de sistema de JMS. Más Información...

Nombre de Archivo Descriptor: jms/xbusResources-jms.xml Nombre del archivo descriptor del módulo JMS. Más Información...

En esta página se resumen los recursos de JMS creados para este módulo de sistema de JMS, incluidos destinos de cola y tema, fábricas de conexiones, plantillas de JMS, claves de clasificación de destinos, cuotas de destinos, destinos distribuidos, servidores ajenos y parámetros de Store-and-Forward.

Personalizar esta Tabla

Resumen de Recursos

Nuevo Suprimir Mostrando 11 a 16 de 16 Anterior Siguiente

<input type="checkbox"/>	Nombre	Tipo	Nombre de JNDI	Despliegue Secundario	Destinos
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2ColaError	Cola	WSLentoPuntoAPunto_PS2ColaError	WSLentoPuntoAPunto_PS2ColaError	wlsbJMSSEver
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Error	Cola	WSLentoPuntoAPunto_PS2Error	WSLentoPuntoAPunto_PS2Error	wlsbJMSSEver
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Petición-980247694	Cola	WSLentoPuntoAPunto_PS2Petición	WSLentoPuntoAPunto_PS2Petición-980247694	wlsbJMSSEver
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Request2080341550	Cola	WSLentoPuntoAPunto_PS2Request	WSLentoPuntoAPunto_PS2Request2080341550	wlsbJMSSEver
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Response-1960145433	Cola	WSLentoPuntoAPunto_PS2Response	WSLentoPuntoAPunto_PS2Response-1960145433	wlsbJMSSEver
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Respuesta-1444563257	Cola	WSLentoPuntoAPunto_PS2Respuesta	WSLentoPuntoAPunto_PS2Respuesta-1444563257	wlsbJMSSEver

Nuevo Suprimir Mostrando 11 a 16 de 16 Anterior Siguiente

Pestaña Supervisión. Seleccionamos la cola y “Mostrar mensajes”.

Valores para WSLentoPuntoAPunto_PS2Respuesta-1444563257

Configuración **Supervisión** Control Seguridad Despliegue Secundario Notas

Un destino de JMS identifica una cola (punto a punto) o un tema (publicación/suscripción) dirigidos a un servidor JMS.

En esta página se resumen los destinos de JMS activos que se han creado para este módulo JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Mostrar Mensajes Mostrando 1 a 1 de 1 Anterior Siguiente

<input checked="" type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Mensajes Totales	Consumidores Actuales	Recuento Alto de Consumidores	Consumidores Totales	Recuento Alto de Mensajes
<input checked="" type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2Respuesta-1444563257	3	0	6	16	16	16	3

Mostrar Mensajes Mostrando 1 a 1 de 1 Anterior Siguiente

Mensajes de JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir Mover Importar Exportar Mostrando 21 a 25 de 25 Anterior Siguiente

<input type="checkbox"/>	Identificador	Identificador de Correlación	Registro de Hora	Cadena de Estado	Modo de Entrega de JMS	Tamaño del Mensaje
<input type="checkbox"/>	ID:<374629.1294821520284.0>	ID:42454153112d310dc016f9360000012d74646172ffff85b3	Wed Jan 12 09:38:40 CET 2011	visible	Persistente	297
<input type="checkbox"/>	ID:<374629.1294821520354.0>	ID:42454153112d310dc016f9360000012d74646172ffff85b6	Wed Jan 12 09:38:40 CET 2011	visible	Persistente	297
<input type="checkbox"/>	ID:<374629.1294822709126.0>	ID:42454153112d310dc016f9360000012d74646172ffff85c3	Wed Jan 12 09:58:29 CET 2011	visible	Persistente	297
<input type="checkbox"/>	ID:<374629.1294822709171.0>	ID:42454153112d310dc016f9360000012d74646172ffff85ca	Wed Jan 12 09:58:29 CET 2011	visible	Persistente	297
<input type="checkbox"/>	ID:<374629.1294822709195.0>	ID:42454153112d310dc016f9360000012d74646172ffff85d1	Wed Jan 12 09:58:29 CET 2011	visible	Persistente	297

Nuevo Suprimir Mover Importar Exportar Mostrando 21 a 25 de 25 Anterior Siguiente

PASO 2

URL WSELETO:

<http://a.b.c.d:8081/WSLento-WSlento-context-root/WSLENTOPort?WSDL>

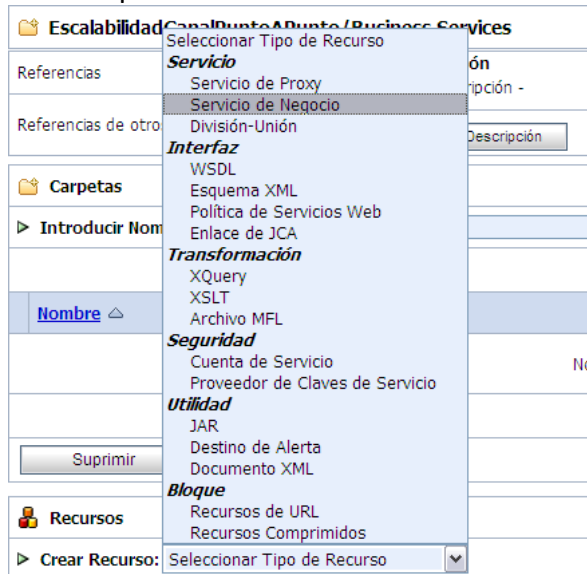
URL WSELETO1:

<http://a.b.c.d:8081/WSLentoooo-WSlento-context-root/WSLENTOPort?WSDL>

Creamos un nuevo proyecto llamado EscalabilidadCanalPuntoAPunto. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

Mismo pasos que en PASO 1 hasta creación del *Business Service*.

En la carpeta *Business Services* creamos un nuevo servicio de negocio.



Nombre del servicio “WSLentoPuntoAPunto_BS” y se crea a partir de cualquier servicio SOAP.

Crear un Servicio de Negocio (EscalabilidadCanalPuntoAPunto/Business Services/)

Configuración General

Nombre del Servicio*: WSLentoPuntoAPunto_NS

Descripción:

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☒ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> | Último >> | Cancelar

“Siguiente”.

Elegimos el Algoritmo de equilibrio de carga “Asignación en rueda”. Añadimos la URL del WSLENTO. Añadimos la URL del WSLENTO1.

En el campo recuento de reintentos ponemos 1, es decir, si la URI a la que accedemos está offline (no disponible) lo reintenta con la siguiente.

Editar un Servicio de Negocio (EscalabilidadServiciosSincronosPaso2/Business Services/WSLento_BS)

Configuración de Transporte

Protocolo*

Algoritmo de Equilibrio de Carga

URI de Punto Final*

Formato:

URI EXISTENTES	OPCIONES
<input type="text" value="http://.../WSLento-WSLento-context-root/WSLENTOPort"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>
<input type="text" value="http://.../WSLentoooo-WSLento-context-root/WSLENTOPort"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>

Recuento de Reintentos

Intervalo de Iteración de Reintentos

Reintentar tras Errores de Aplicación ☒ Sí ☐ No

“Último” y “Guardar”.

Seguir con los mismos pasos que en PASO 1.

Prueba

URL EscalabilidadCanalPuntoAPunto OSB:

http://a.b.c.d:8081/EscalabilidadCanalPuntoAPunto/Proxy_Services/WSLentoPuntoAPunto_PS1?wsdl

La aplicación se puede probar desde:

<http://a.b.c.d:8081/CienteEscalabilidadPuntoAPunto/index.jsp>

Si ejecutamos la aplicación obtenemos el instante de la llamada por pantalla:

Escalabilidad en canal punto a punto

Minuto inicio llamadas 24 Segundo inicio llamadas 41

En la consola del OSB se puede llevar un seguimiento de las llamadas.

Accedemos a Operaciones / Estado de servicio.

Seleccionamos el servicio de negocio creado WSLentoPuntoAPunto_BS.

ORACLE® Service Bus 11gR1

Bienvenido, weblogic Conectado a: poc_osbsoa Inicio Consola de Oracle V

Alertas de SLA Alertas de Pipeline **Estado de Servicio** Estado del Servidor 1

Estado de Servicio

Mostrar Estadísticas Intervalo de Agregación Actual

Servidor osb_server1

Búsqueda

Nombre

Ruta de Acceso

Filtros de Estado de Servicio

Búsqueda Ver Todo

Nombre	Ruta de Acceso	Tipo de Servicio	Intervalo de Agregación	Tiempo Medio de Respuesta	Me
Servicio de Seguimiento	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio de Seguimiento (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_BS	EscalabilidadCanalPuntoAPunto/Business S...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_BS	EscalabilidadServiciosPuntoAPuntoPaso1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_PS2	EscalabilidadCanalPuntoAPunto/Proxy Serv...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
WSLentoPuntoAPunto_PS2	EscalabilidadServiciosPuntoAPuntoPaso1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	

Principio

Pestaña Métricas de servicio. Comprobamos que llegan 3 mensajes:

Métricas de Servicio URI de Punto Final

General

Recuento de Alertas de SLA	Alertas de 0 ; Normal 12/01/11 10:26
Tiempo Mínimo de Respuesta	30 segs. 6 msecs.
Tiempo Máximo de Respuesta	30 segs. 9 msecs.
Tiempo Medio de Respuesta General	30 segs. 7 msecs.
Recuento de Mensajes	3
Recuento de Errores	0
Recuento de Failover	0
Ratio de Éxitos	100,00%
Ratio de Fallos	0,00%
Errores de WS-Security	0
Errores de Validación	N/A

Almacenamiento en Caché de Resultados

Recuento de Aciertos	0
Ratio de Aciertos	0,00%

Regulación

Tiempo Mínimo de Regulación	0 msecs.
Tiempo Máximo de Regulación	0 msecs.
Tiempo Medio de Regulación	0 msecs.

Pestaña URI de Punto Final. Comprobamos el reparto de los mensajes.

Métricas de Servicio URI de Punto Final

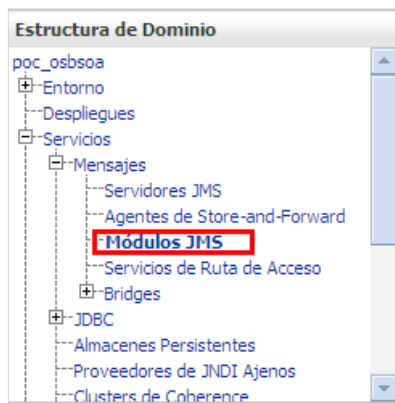
Elementos 1 - 2 de

URI de Punto Final	Recuento de Mensajes	Recuento de Errores	Tiempo Mínimo de Respuesta	Tiempo Máximo de Respuesta	Tiempo Medio de Respu
http://10.238.7.6:8011/WSLento-WSLento-context-root/WSL...	2	0	30 segs. 6 msecs.	30 segs. 9 msecs.	30 segs. 7 msecs.
http://10.238.7.6:8011/WSLento0000-WSLento-context-root/...	1	0	30 segs. 8 msecs.	30 segs. 8 msecs.	30 segs. 8 msecs.

Elementos 1 - 2 de

Atrás Restablecer Estadística Refrescar

Para ver el momento en que llegan los mensajes vamos a Weblogic, Servicios / Módulos JMS.



Módulo JMS jmsResources.

Módulos JMS

Los recursos del sistema de JMS se configuran y almacenan como módulos parecidos a los módulos de J2EE estándar. Dichos recursos incluyen colas, temas, fábricas de conexiones, plantillas, claves de destino, cuotas, colas distribuidas, temas distribuidos, servidores ajenos y parámetros de Store-and-Forward (SAF) de JMS. Puede configurar y gestionar de forma administrativa módulos de sistema de JMS como recursos del sistema globales.

En esta página se resumen los módulos del sistema de JMS creados para este dominio.

[Personalizar esta Tabla](#)

Módulos JMS

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Tipo
<input type="checkbox"/>	BPMJMSModule	Sistema
<input type="checkbox"/>	jmsResources	Sistema
<input type="checkbox"/>	SOAJMSModule	Sistema
<input type="checkbox"/>	SystemModule	Sistema
<input type="checkbox"/>	UMSJMSSystemResource	Sistema
<input type="checkbox"/>	WseeJmsModule	Sistema

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

Seleccionamos WSLentoPuntoAPunto_PS2Resonse.

Valores para jmsResources

[Configuración](#) [Despliegues Secundarios](#) [Destinos](#) [Seguridad](#) [Notas](#)

Esta página muestra información general sobre un módulo de sistema de JMS y sus recursos. También permite configurar nuevos recursos y acceder a los recursos existentes.

Nombre: jmsResources [Nombre de este módulo de sistema de JMS. Más Información...](#)

Nombre de Archivo Descriptor: jms/xbusResources-jms.xml [Nombre del archivo descriptor del módulo JMS. Más Información...](#)

En esta página se resumen los recursos de JMS creados para este módulo de sistema de JMS, incluidos destinos de cola y tema, fábricas de conexiones, plantillas de JMS, claves de clasificación de destinos, cuotas de destinos, destinos distribuidos, servidores ajenos y parámetros de Store-and-Forward.

[Personalizar esta Tabla](#)

Resumen de Recursos

[Nuevo](#) [Suprimir](#) Mostrando 21 a 21 de 21 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Tipo	Nombre de JNDI	Despliegue Secundario	Destinos
<input type="checkbox"/>	WSLentoPuntoAPunto_PS2Response508074449	Cola	WSLentoPuntoAPunto_PS2Response	WSLentoPuntoAPunto_PS2Response508074449	wlsbJMServer

[Nuevo](#) [Suprimir](#) Mostrando 21 a 21 de 21 Anterior | Siguiente

Pestaña Supervisión. Seleccionamos la cola y “Mostrar mensajes”.

Valores para WSLentoPuntoAPunto_PS2Response-2003582498

Configuración **Supervisión** Control Seguridad Despliegue Secundario Notas

Un destino de JMS identifica una cola (punto a punto) o un tema (publicación/suscripción) dirigidos a un servidor JMS.
En esta página se resumen los destinos de JMS activos que se han creado para este módulo JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Mostrar Mensajes

Mostrando 1 a 1 de 1 Anterior | Siguiente

<input checked="" type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Mensajes Totales	Consumidores Actuales	Recuento Alto de Consumidores	Consumidores Totales	Recuento Alto de Mensajes
<input checked="" type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2Response- 2003582498	3	0	5	16	16	16	3

Mostrar Mensajes

Mostrando 1 a 1 de 1 Anterior | Siguiente

Mensajes de JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir Mover Importar Exportar

Mostrando 1 a 3 de 3 Anterior | Siguiente

<input type="checkbox"/>	Identificador	Identificador de Correlación	Registro de Hora	Cadena de Estado	Modo de Entrega de JMS	Tamaño del Mensaje
<input type="checkbox"/>	ID: <374629.1294824311422.0>	ID:42454153112d310dc016f9360000012d74646172ffff85da	Wed Jan 12 CET 2011 10:25:11	visible	Persistente	297
<input type="checkbox"/>	ID: <374629.1294824311426.0>	ID:42454153112d310dc016f9360000012d74646172ffff85dd	Wed Jan 12 CET 2011 10:25:11	visible	Persistente	297
<input type="checkbox"/>	ID: <374629.1294824311429.0>	ID:42454153112d310dc016f9360000012d74646172ffff85e0	Wed Jan 12 CET 2011 10:25:11	visible	Persistente	297

Nuevo Suprimir Mover Importar Exportar

Mostrando 1 a 3 de 3 Anterior | Siguiente

Si paramos los dos *Web Services* desplegados en Weblogic y lanzamos la aplicación, la petición (el mensaje) se guarda en la cola de errores JMS.

Despliegues

Instalar Actualizar Suprimir Iniciar Parar

Mostrando 11 a 20 de 166 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Estado	Tipo	Orden de Despliegue
<input type="checkbox"/>	AAWSBasadoContenido-WSService	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSDescripcion	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSDescripcionCompleta	Activo	Aplicación Web	100
<input checked="" type="checkbox"/>	AAWSLento-WSLento-context-root	Activo	Aplicación Web	100
<input checked="" type="checkbox"/>	AAWSLentoooo-WSLento-context-root	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSOrdenDePago	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSRegistro	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSRegistro2	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSRegistro4	Activo	Aplicación Web	100
<input type="checkbox"/>	AAWSsample1-Project1-context-root	Activo	Aplicación Web	100

Instalar Actualizar Suprimir Iniciar Parar

Mostrando 11 a 20 de 166 Anterior | Siguiente

Cola de errores:

Destinos (Filtrado: Existen Más Columnas)

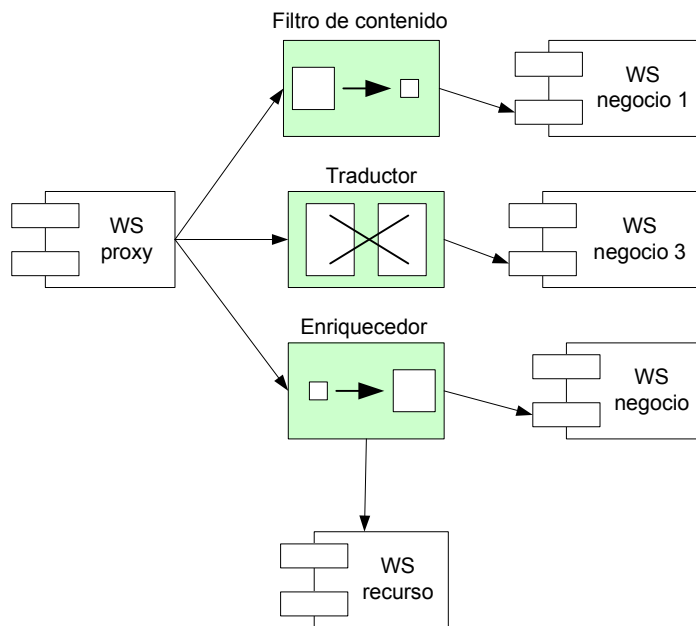
Producción ▼ Consumo ▼ Inserción ▼			Mostrando 1 a 10 de 14 Anterior Siguiente								
<input type="checkbox"/>	Nombre ↕	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Co en Pa
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2ColaError	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2Error	1	0	1	1	0	Queue	advertised_in_cluster_jndi	false	false	fals
<input type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2Petición563670049	0	0	13	24	0	Queue	advertised_in_cluster_jndi	false	false	fals
Producción ▼ Consumo ▼ Inserción ▼			Mostrando 1 a 10 de 14 Anterior Siguiente								

Si paramos un único *Web Services*, todas las peticiones serán tratadas por el *Web Services* que queda activo.

Caso de Uso 4: Transformación de mensajes

Alcance:

El objetivo es demostrar la transformación de mensajes que proporciona el bus. Tendremos tres servicios de negocio y un servicio proxy. El primer servicio de negocio espera un mensaje cuyo contenido sea parte del original. El segundo, espera un mensaje con el mismo contenido pero de manera diferente (otro formato, distinto número de campos). El tercero espera un mensaje cuyo contenido sea el original enriquecido por otro servicio auxiliar.



En este ejemplo el Proxy hace referencia al wsdl FormatoDatos.wsdl

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type*

Create a New Service

☒ WSDL Web Service

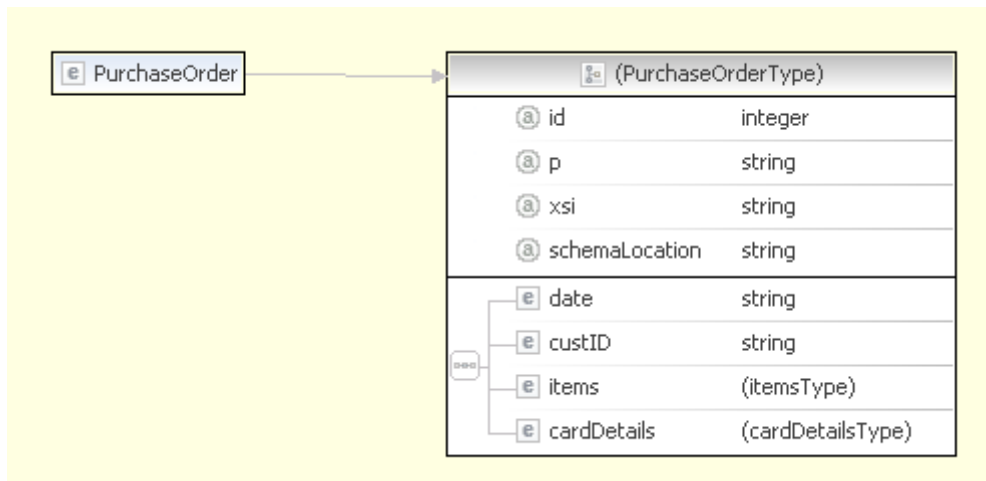
☐ Transport Typed Service

☐ Messaging Service

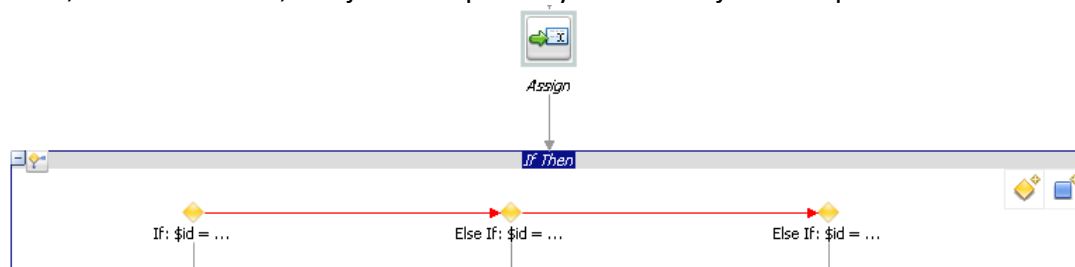
☐ Any SOAP Service

☐ Any XML Service

Los datos que se piden como entrada son:



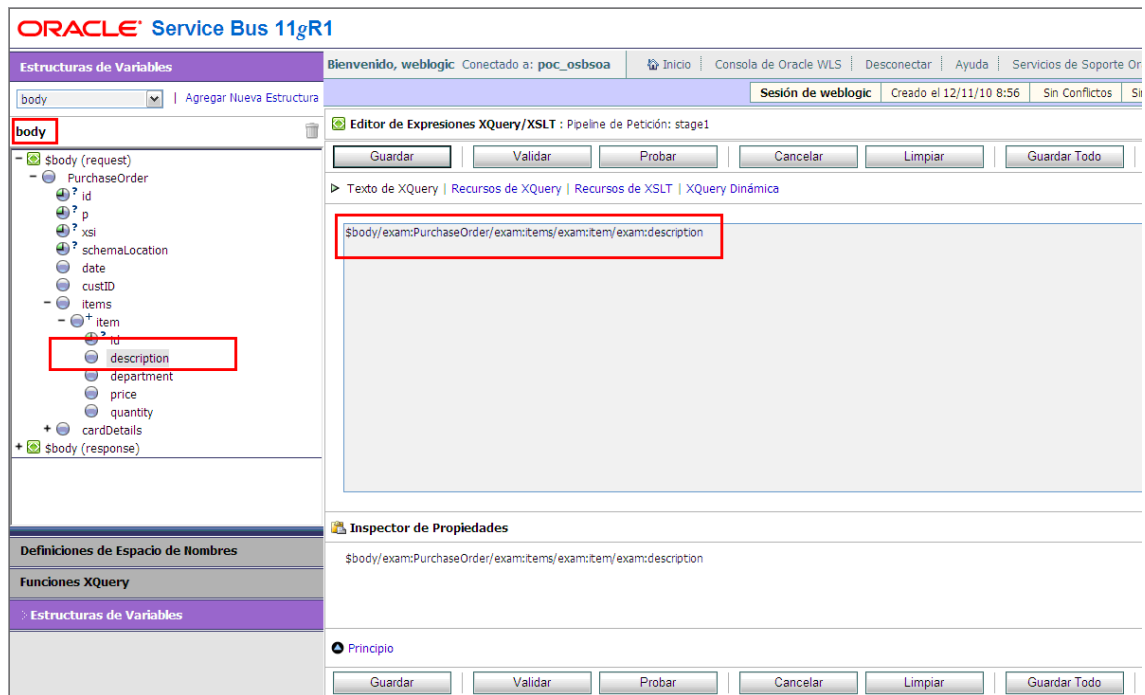
Uno de los datos que se piden en la entrada es custID. Si el custID es 1 se ejecuta el paso 1, si el custID es 2, se ejecuta el paso 2 y si es 3 se ejecuta el paso 3.



PASO 1

El primer servicio de negocio espera un mensaje cuyo contenido sea parte del original. Ejemplo: metemos un conjunto de ítems con varios campos y solo nos interesa pasarle al servicio de negocio el campo descripción de cada uno de los campos. Si sólo tuviéramos un único ítem, es decir, nos sería iterativo, utilizaríamos una consulta XPath como por ejemplo:

`$body/exam:PurchaseOrder/exam:items/exam:item/exam:description`



En este caso vamos a tener varios registros por lo que necesitamos utilizar XQuery o XSLT.

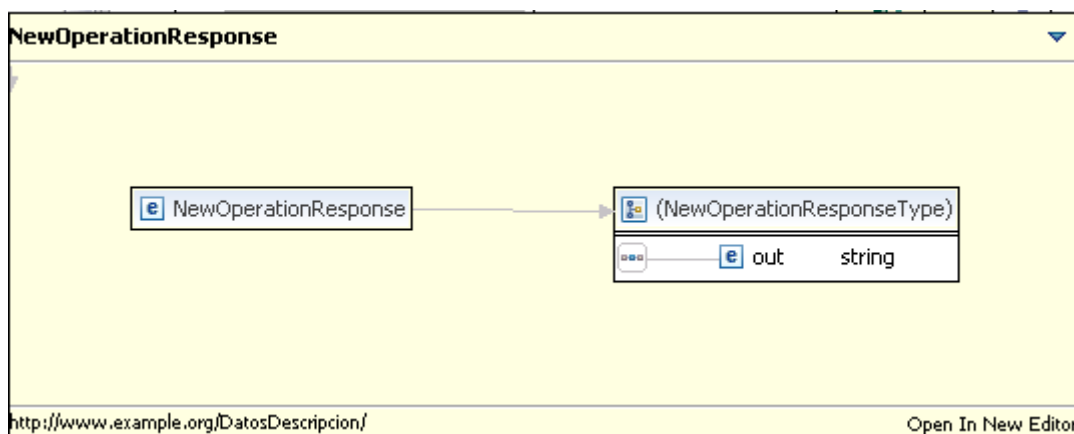
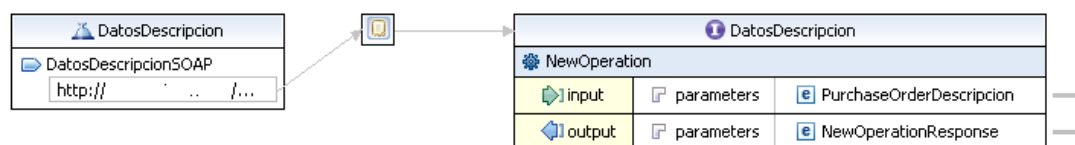
El servicio al que queremos hacer referencia sigue el siguiente esquema XML (items2.xsd):



Queremos el campo descripción de cada uno de los ítems.

Creamos el *Business Service* al que queremos hacer referencia. Para ello añadimos el wsdl y creamos el *Business Service* a partir de él.

(DatosDescripcionSOAP.wsdl)



(DatosDescripcion.biz)

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type *

Create a New Service

☒ WSDL Web Service TransformacionDeMensajes/Recursos/DatosDescripcionSOAP

SOAP 1.1 DatosDescripcionSOAP (port)

☐ Transport Typed Service

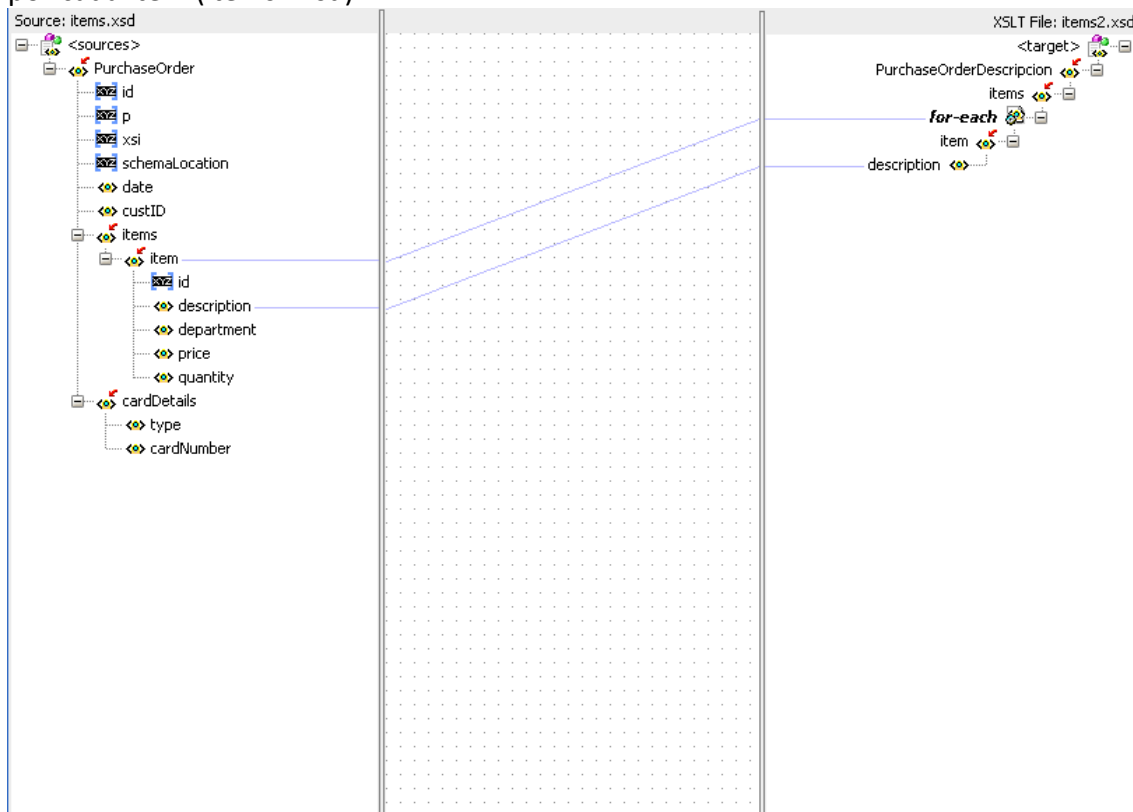
☐ Messaging Service

☐ Any SOAP Service SOAP 1.1

☐ Any XML Service

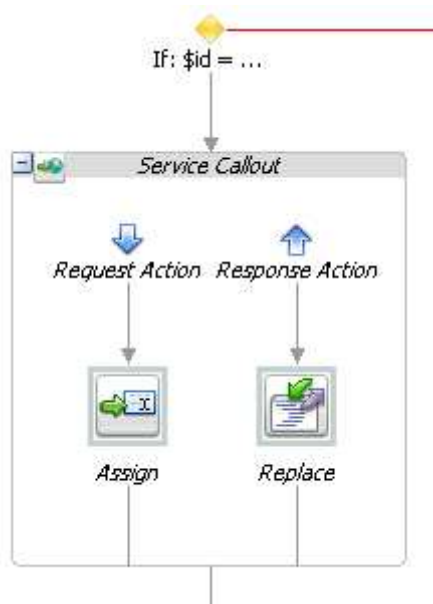
A continuación procedemos a la transformación del mensaje. Queremos transformar el esquema de entrada al esquema items2.xsd que hemos creado. Para ello creamos un esquema XSLT (también se puede hacer con transformación XQuery). (archivo DatosDescripcion.xsl)

Mediante este esquema XSLT transformamos el mensaje de entrada en el mensaje de salida, es decir, cogemos de los datos de entrada (ítem.xsd) solo el campo descripción por cada ítem (items2.xsd).

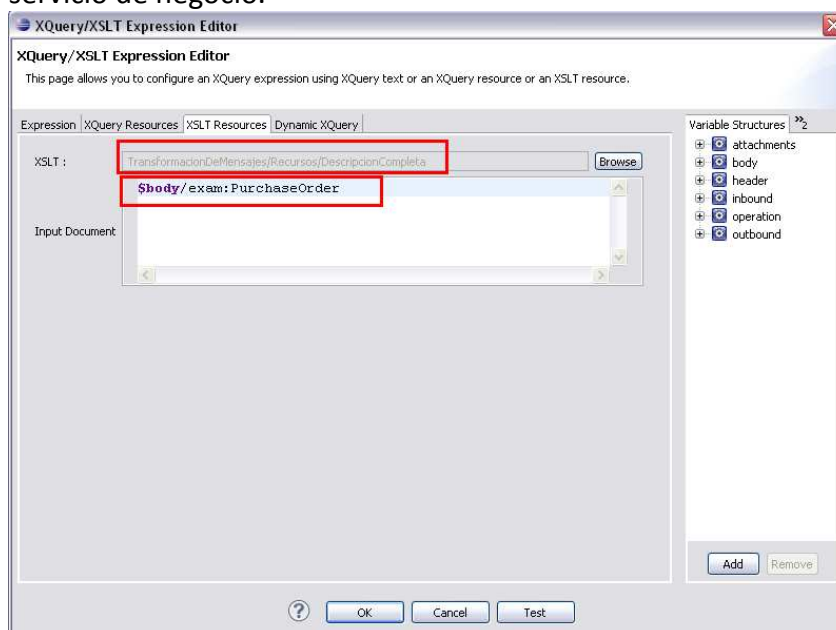


Para crear el XSLT se ha utilizado Jdeveloper.

Una vez que hemos creado todos los recursos, modificamos el flujo de mensajes del proxy:



En el “assign” hacemos una llamada al XSLT que hemos creado pasándole el mensaje de entrada y asignamos el resultado a la variable que se le pasa como entrada al servicio de negocio.

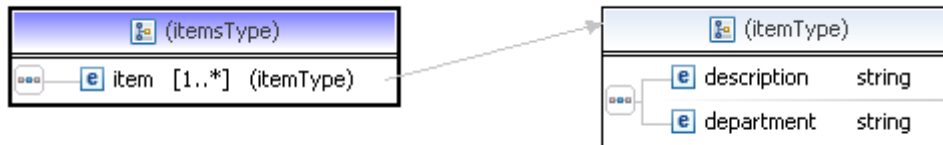


En el “replace” remplazamos el body por el mensaje devuelto por el servicio de negocio.

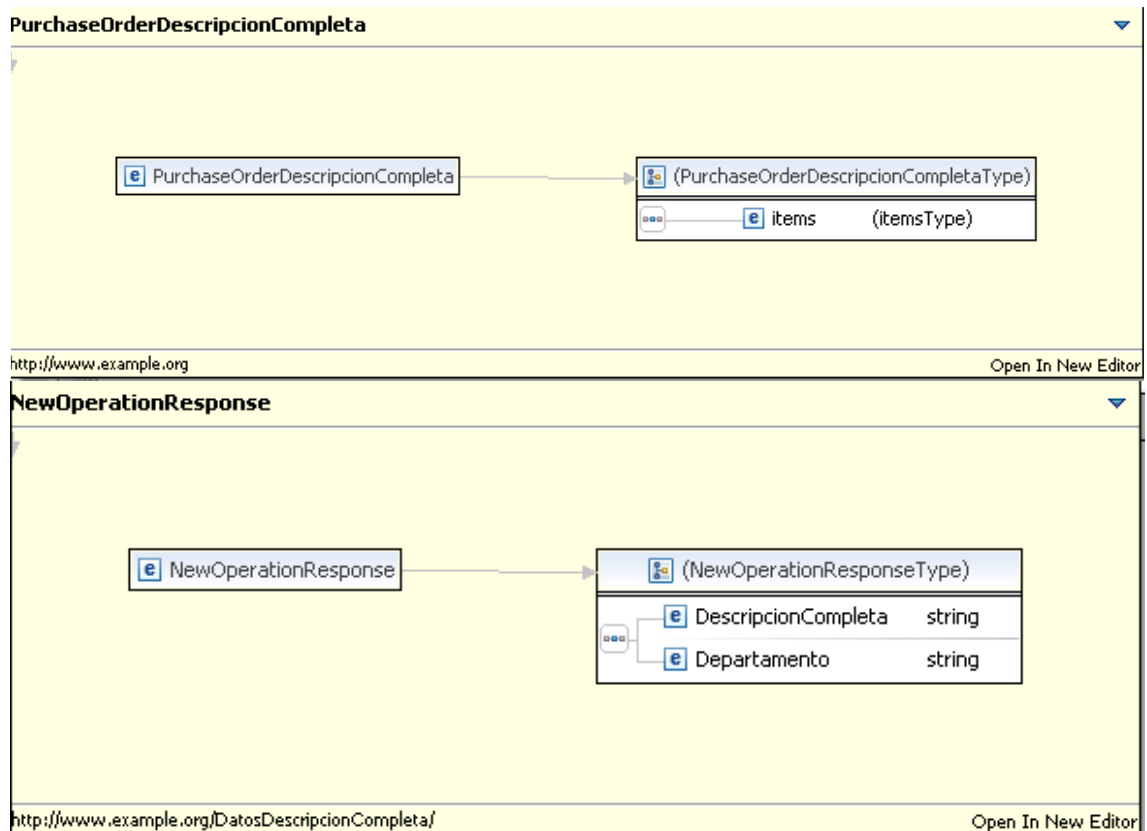
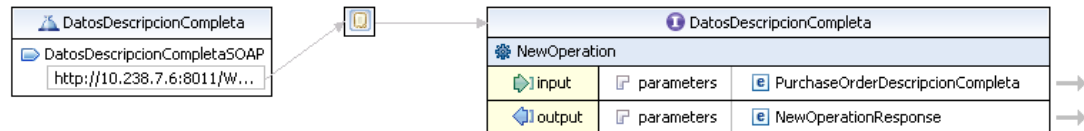
PASO 2

En este ejemplo queremos mostrar el contenido de los campos descripción y precio en un único campo descripción, dichos contenidos separados por un guión y el campo departamento en mayúsculas (cambio de campos y formatos).

El servicio al que queremos hacer referencia sigue el siguiente esquema XML (items3.xsd):



Creamos el *Business Service* al que queremos hacer referencia. Para ello añadimos el wsdl y creamos el *Business Service* a partir de él.
(DatosDescripcionCompletaSOAP.wsdl)



(DatosDescripcionCompleta.biz)

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type *

Create a New Service

☒ WSDL Web Service

SOAP 1.1

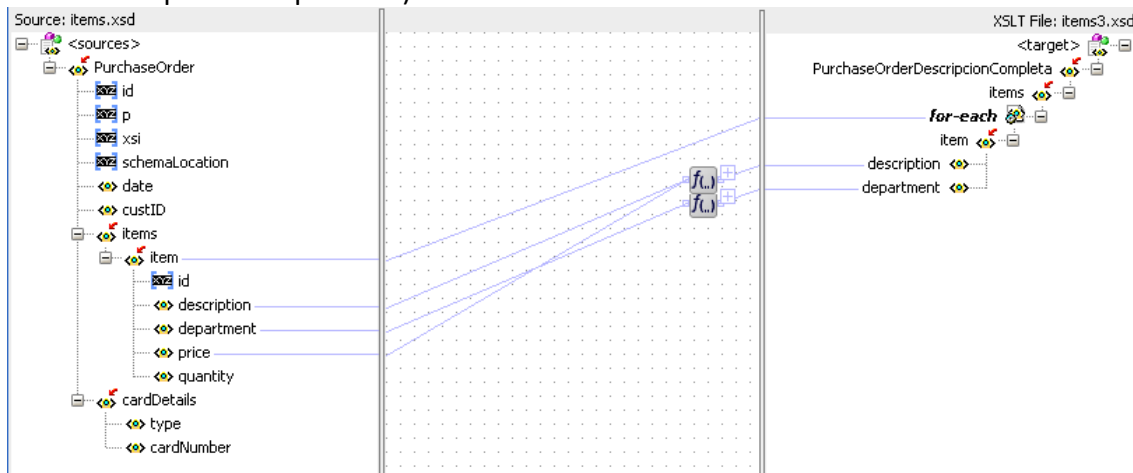
☐ Transport Typed Service

☐ Messaging Service

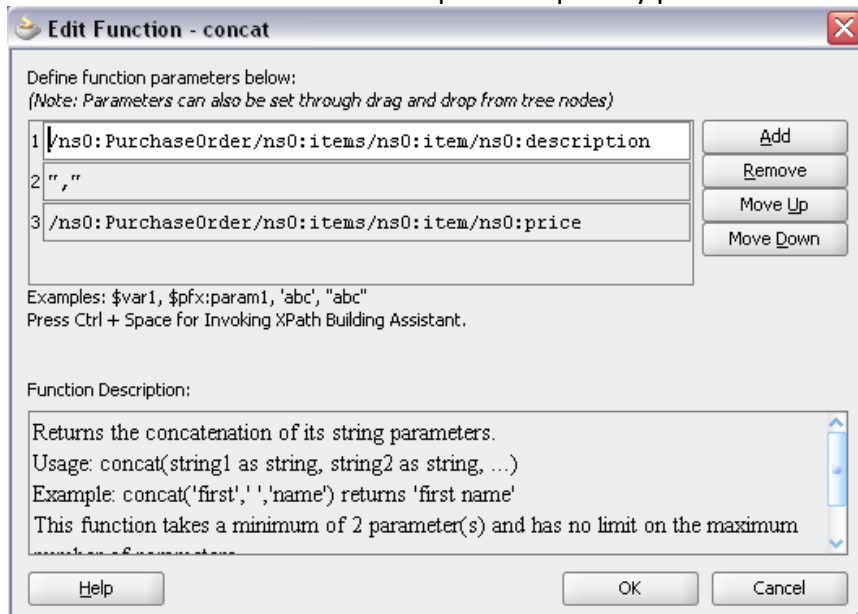
☐ Any SOAP Service

☐ Any XML Service

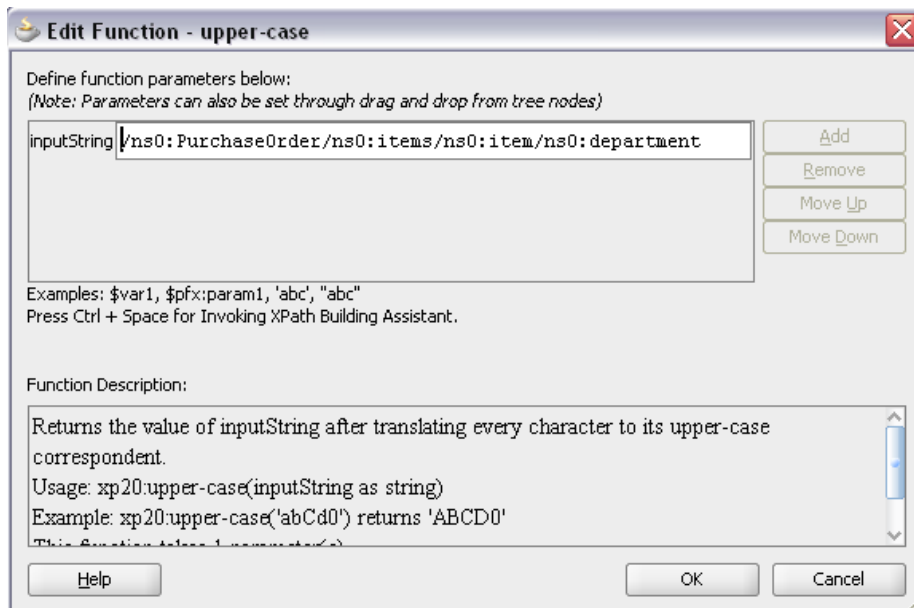
A continuación procedemos a la transformación del mensaje. Queremos transformar el esquema de entrada al esquema items3.xsd que hemos creado. Para ello creamos un esquema XSLT (también se puede hacer con transformación XQuery). (archivo DatosDescripcionCompleta.xsl)



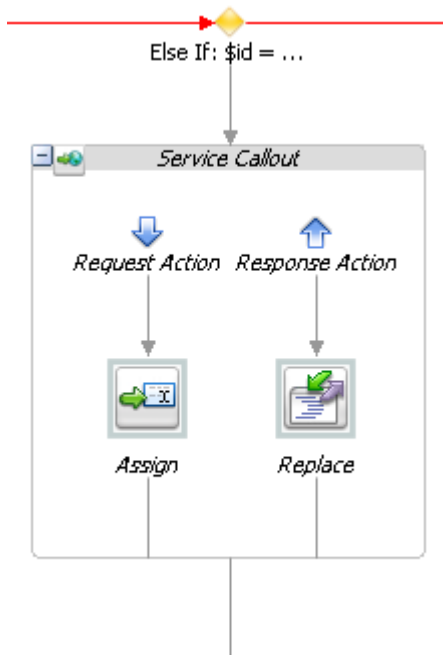
Procedemos a concatenar el campo descripción y precio:



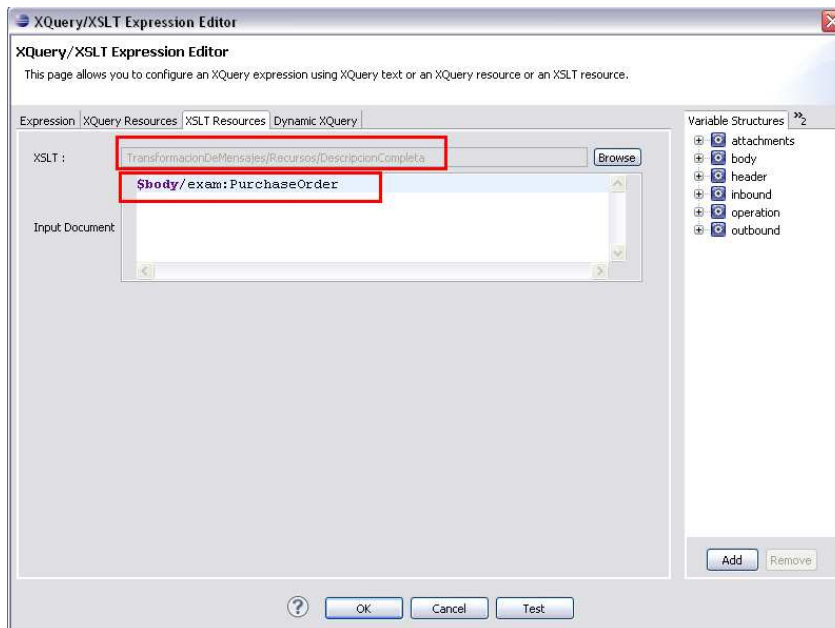
Procedemos a cambiar el campo departamento a mayúsculas:



Una vez que hemos creado todos los recursos, modificamos el flujo de mensajes del proxy:



En el “assign” hacemos una llamada al XSLT que hemos creado pasándole el mensaje de entrada y asignamos el resultado a la variable que se le pasa como entrada al servicio de negocio.



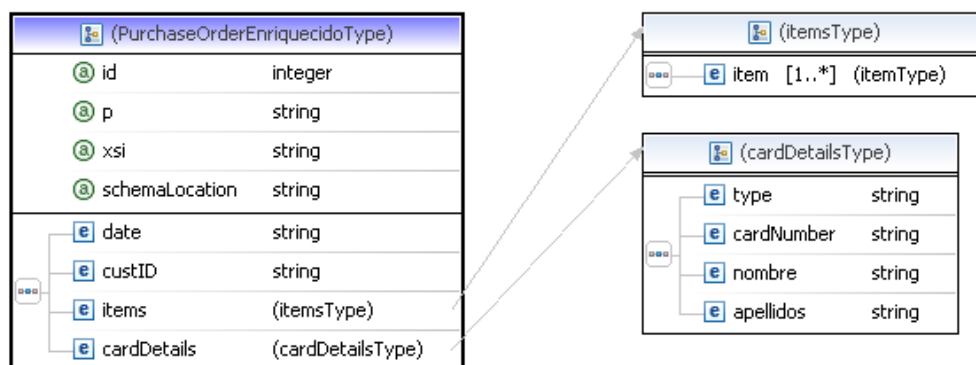
En el “replace” reemplazamos el body por el mensaje devuelto por el servicio de negocio.

PASO 3

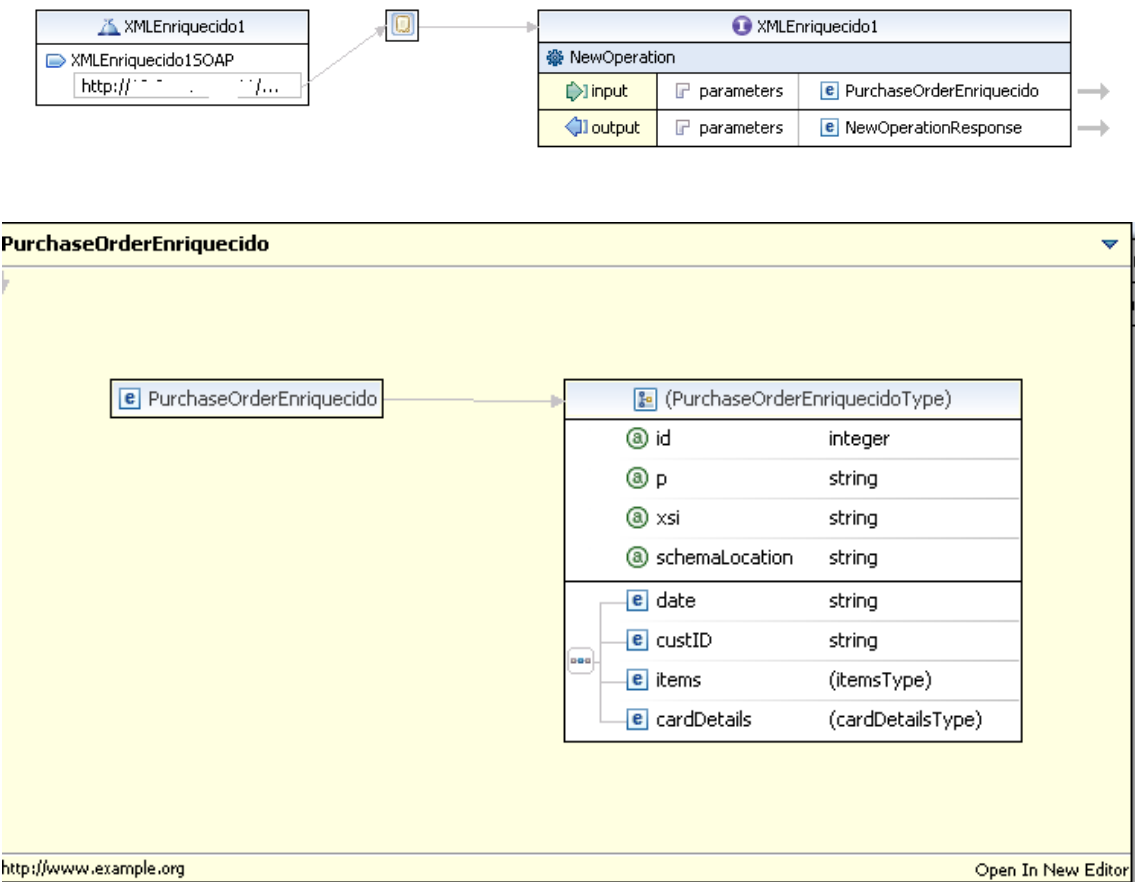
El tercero espera un mensaje cuyo contenido sea el original enriquecido por otro servicio auxiliar. Es decir, mostramos los campos iniciales y le sumamos dos nuevos campos, nombre y apellidos, que se obtienen de la llamada a otro servicio de negocio. Para este caso hemos utilizado Split Join.

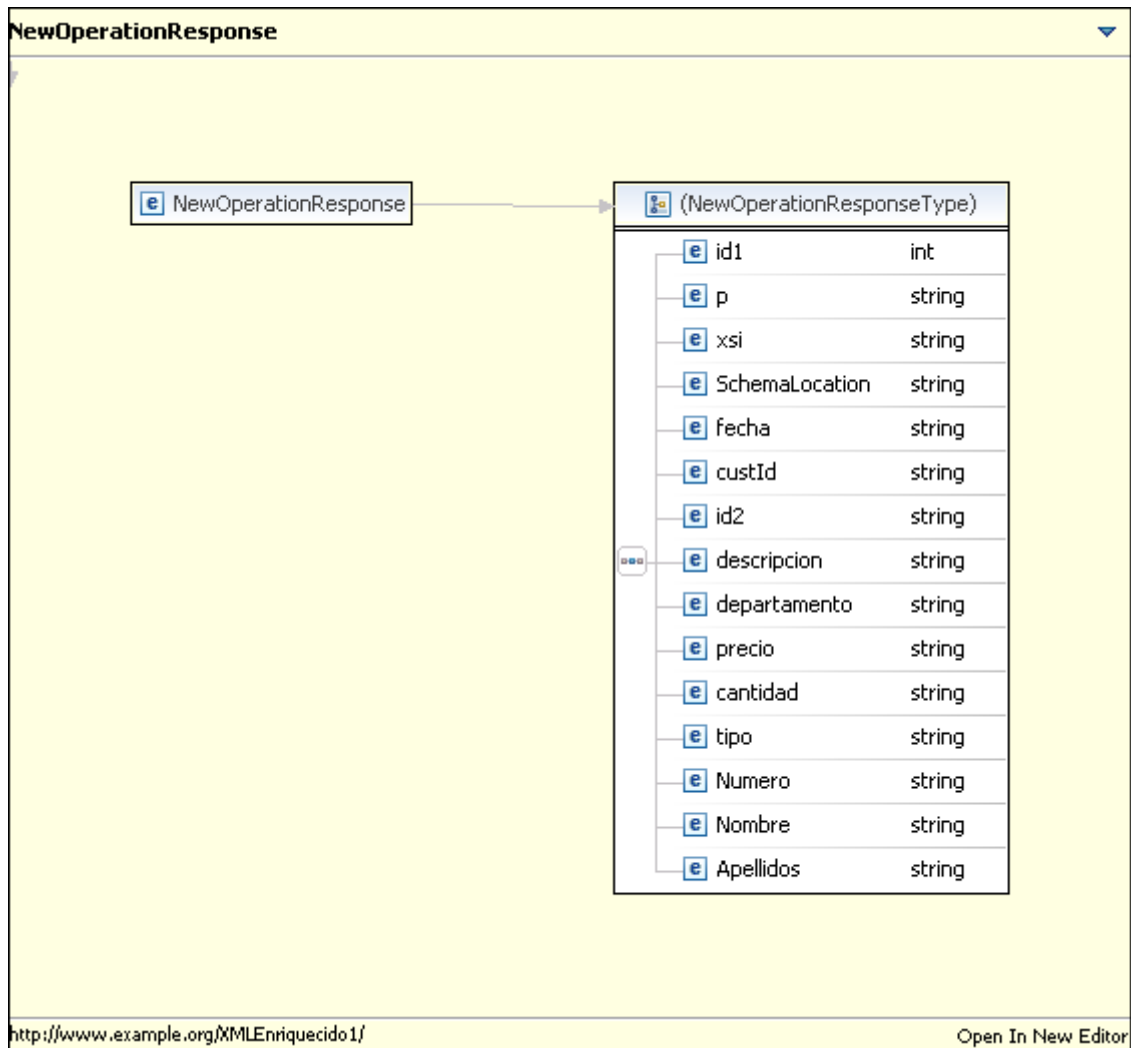
Creamos el WSDL del *Business Service* al que vamos a querer llamar, XMLEnriquecido1SOAP.wsdl, es decir el *Business Service* cuyo contenido sea el original enriquecido por otro servicio auxiliar.

Estructura de datos: (Items4.xsd)



WSDL:





Creamos el servicio de negocio a partir de este WSDL que hemos creado (XMLEnriquecido.biz)

The screenshot shows the 'General Configuration' window for the 'XMLEnriquecido.biz' service. The window has a tab bar at the top with 'AnadirDatos.biz', 'DatosDescripcion.biz', 'SplitJoin.biz', and 'XMLEnriquecido.biz'. The 'Configuration' tab is active, showing a description field and a 'Service Type' section. The 'Service Type' section has a 'Create a New Service' button and a list of service types: 'WSDL Web Service' (selected), 'Transport Typed Service', 'Messaging Service', 'Any SOAP Service', and 'Any XML Service'. The 'WSDL Web Service' type is configured with the URL 'TransformacionDeMensajes/Recursos/XMLEnriquecido1SOAP' and the SOAP version 'SOAP 1.1'.

Añadimos un nuevo WSDL llamado AnadirDatos. Este será el servicio auxiliar. Tiene dos operación: AndarApellidos y AnadirNombre.

Una vez añadimos creamos su respectivo *Business Services*: (AnadirDatos.biz)

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type *

Create a New Service

☒ WSDL Web Service

SOAP 1.1 (port)

☐ Transport Typed Service

☐ Messaging Service

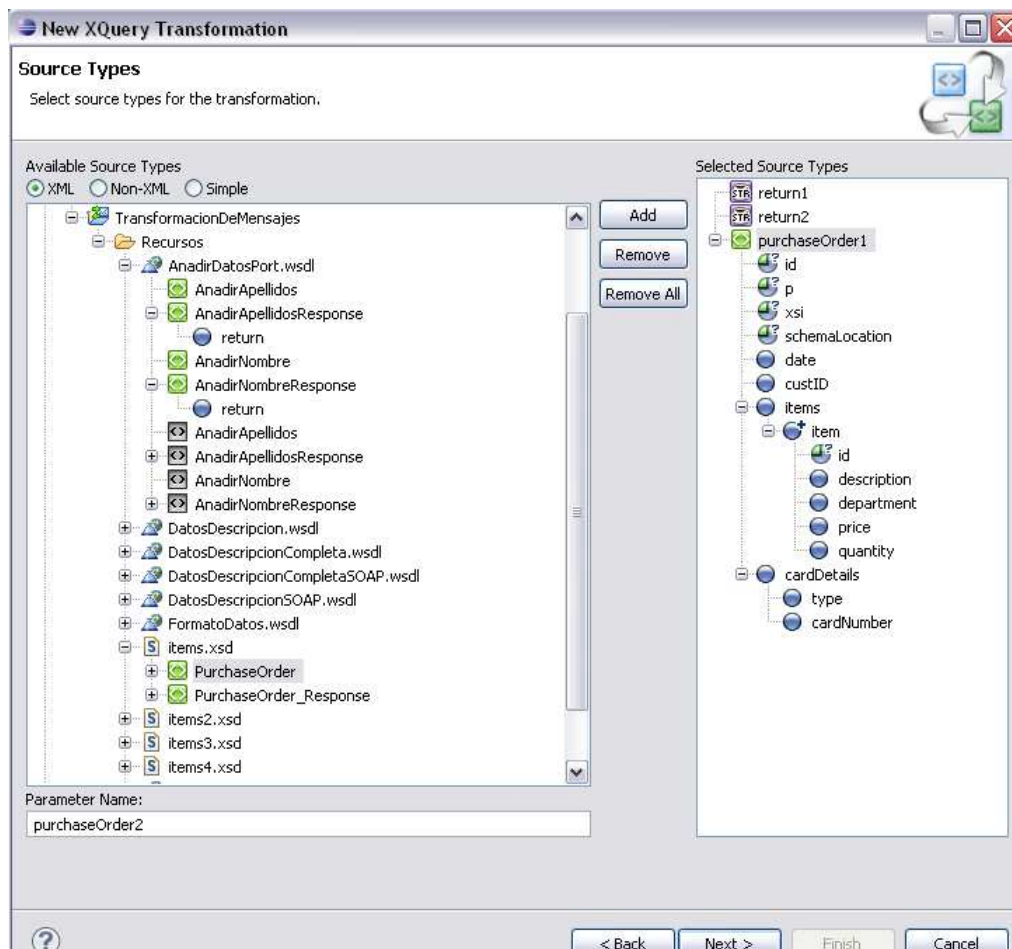
☐ Any SOAP Service

☐ Any XML Service

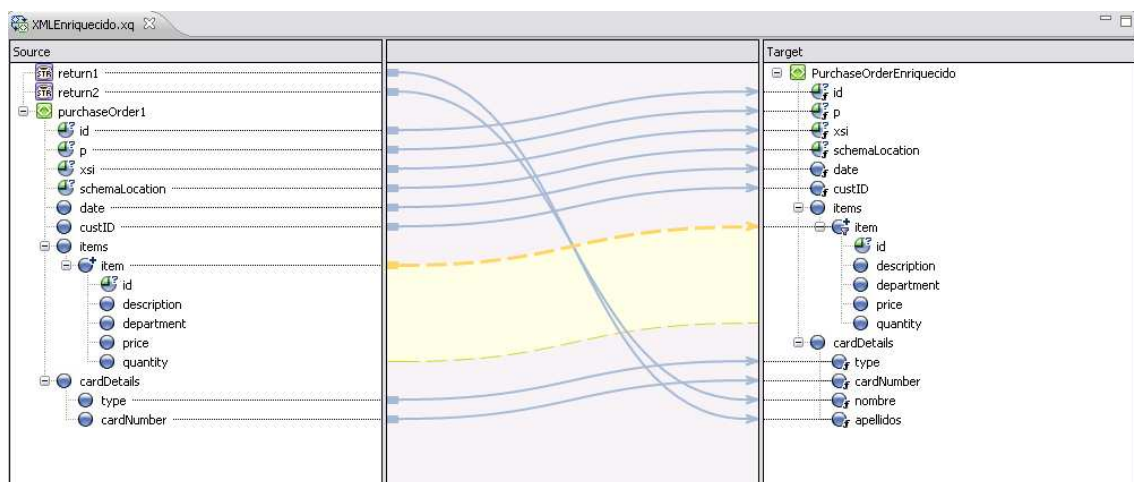
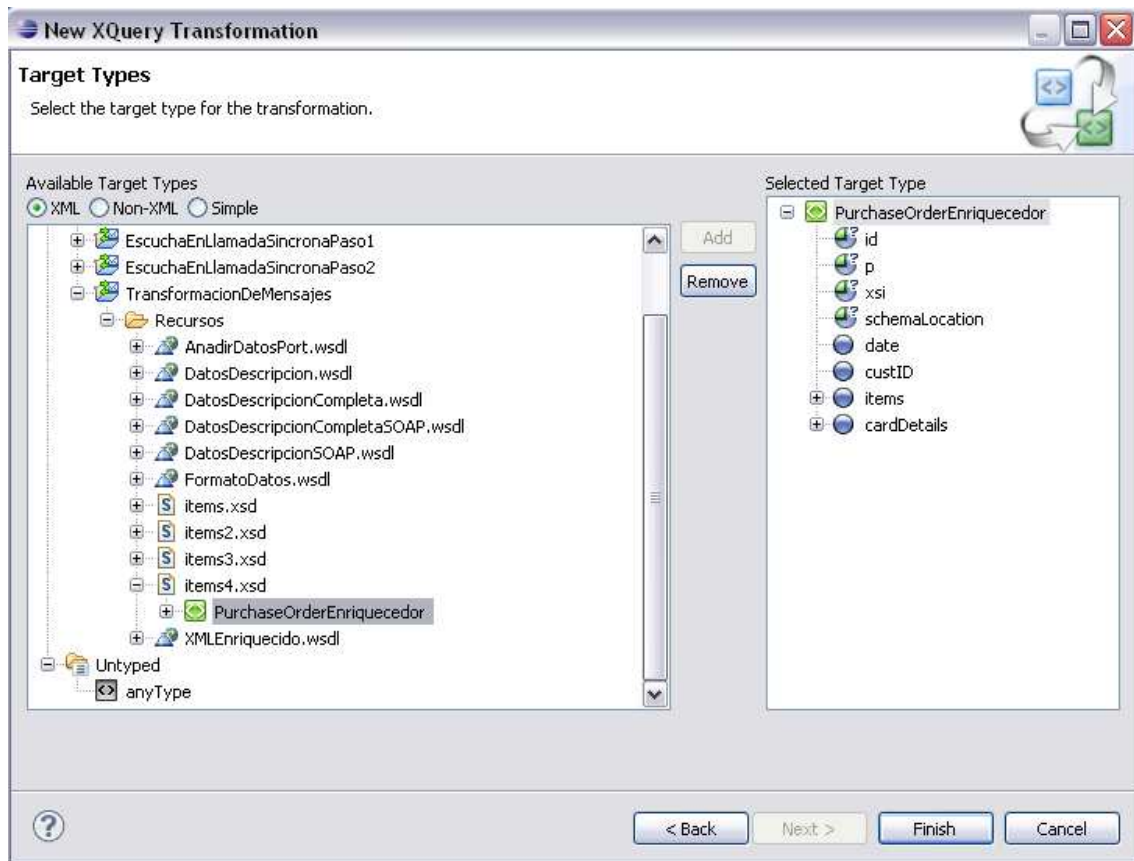
En el Split join (SplitJoin.flow) lo que hacemos es llamar de forma paralela a dos métodos diferentes de un *Business Service* (AnadirDatos.biz). De manera que en la variable \$nombre guardamos lo que nos devuelve una de las llamadas y en \$apellidos lo que nos devuelve la otra llamada. Una vez que tenemos estas dos variables creamos una consulta XQuery para asignar los resultados obtenidos a la respuesta esperada por el servicio de negocio:

(XMLEnriquecido.xq)

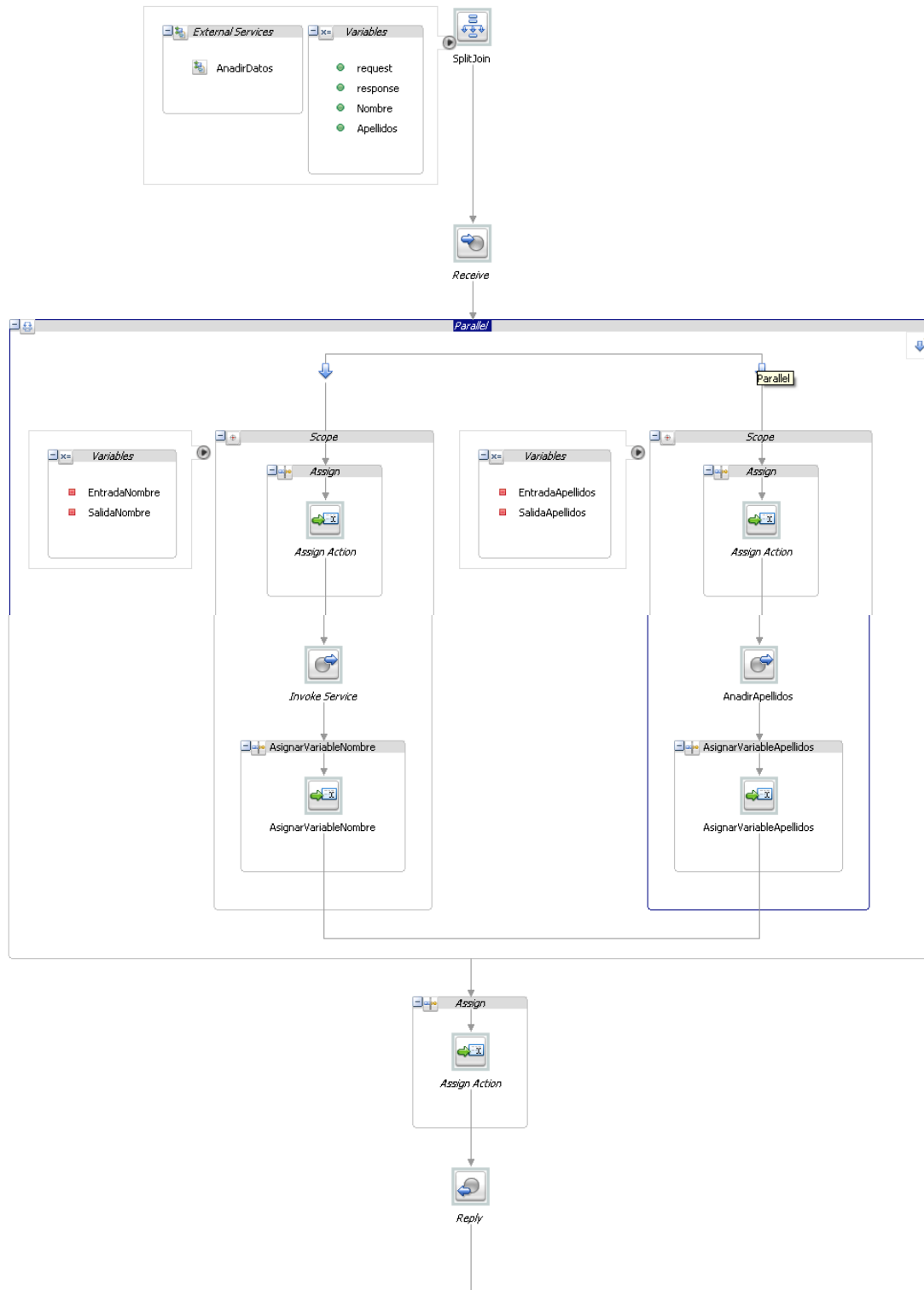
- Como entrada tenemos: el return del método AnadirApellidosResponse (en este caso \$apellidos), el return del método AnadirNombreResponse(en este caso \$nombre) y el mensaje de entrada (PurchaseOrder).



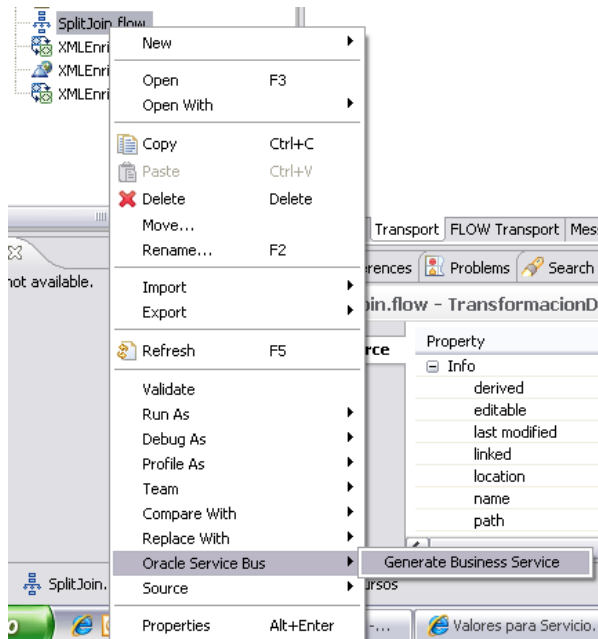
- Como salida tenemos: PurchaseOrderEnriquecedor, es decir, el mensaje original enriquecido por el nombre y apellidos.



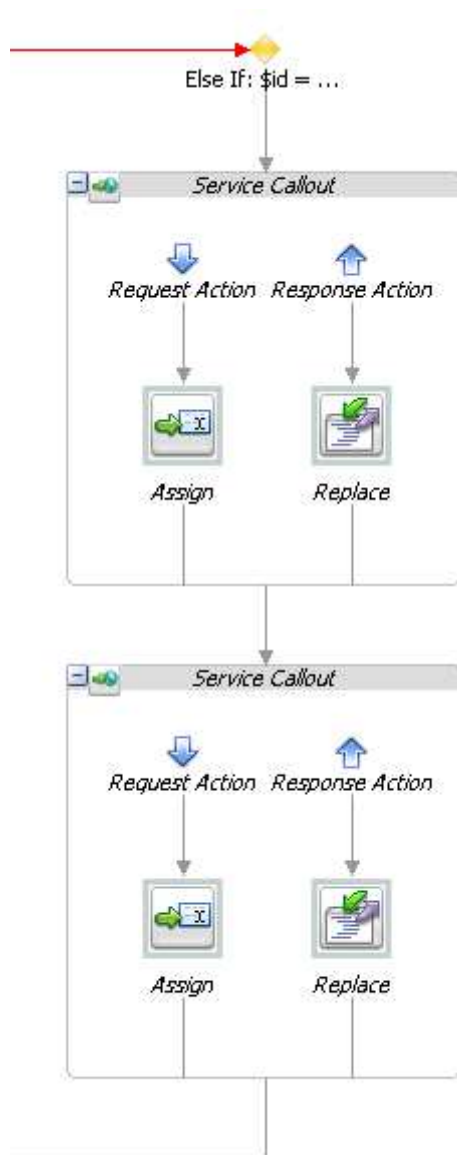
Una vez creada la transformación XQuery se la asignamos al campo expresión del último “Assign Action” y se lo asignamos a la variable de respuesta (response). De esta manera ya tenemos el mensaje enriquecido.



Creamos el *Business Services* de este Split join de la siguiente forma: (SplitJoin.biz)



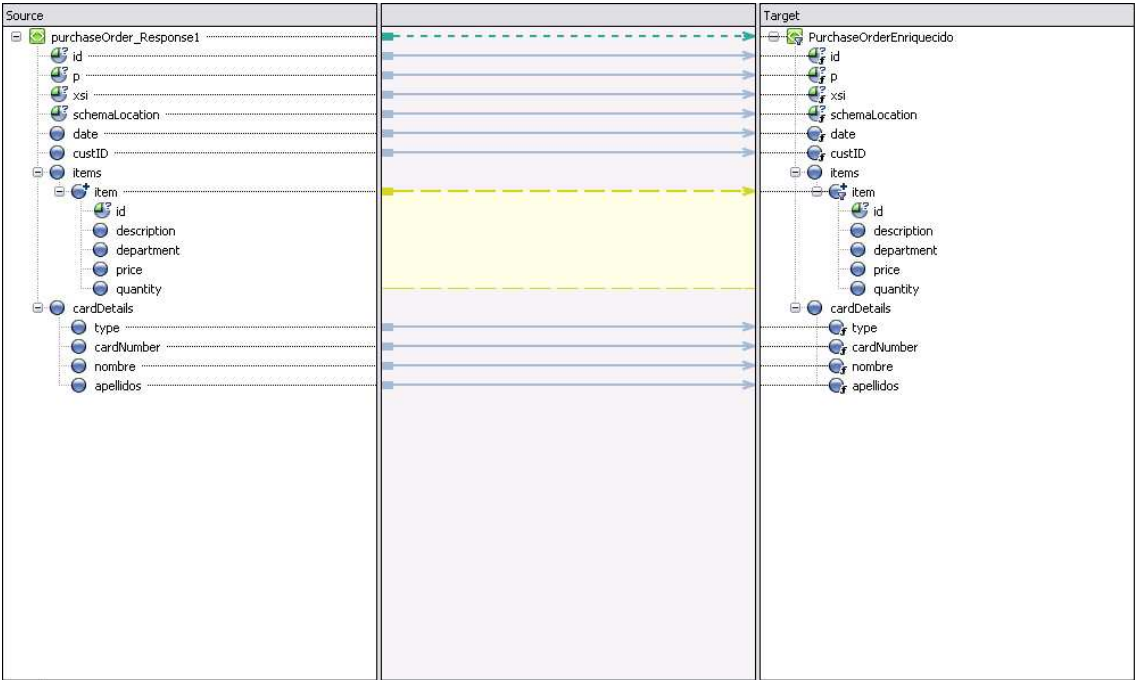
En el *Proxy Service* (Regitroltems.proxy): añadimos dos “service callout”. El primero se va a encargar de enriquecer el mensaje y el mensaje enriquecido será pasado al segundo “service callout”.



En el primero llama a SplitJoin.biz, asignamos como entrada el mensaje principal y como salida el mensaje enriquecido.

Una vez tenemos el mensaje enriquecido se lo asignamos como entrada al siguiente "service callout" que llama al *Business Service XMLEnriquecido.biz*.

La salida del primer "service callout" y la entrada del segundo no tienen la misma estructura de datos, por lo que hacemos la siguiente transformación XQuery: (XMLEnriquecido2.xq)

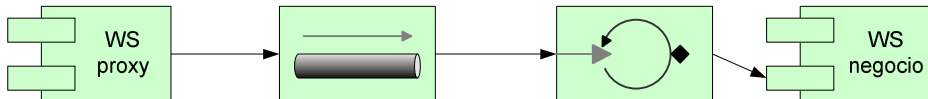


CASOS DE USO DE SPE

Caso de Uso 1: Consumidor de mensajes (Polling consumer)

Alcance:

El objetivo es demostrar la implementación de un consumidor de mensajes que actúe de intermediario entre un servicio Web de un solo sentido y una cola de mensajes.



Tendremos un servicio de negocio SOAP síncrono de un solo sentido y una aplicación que llamará a un proxy que también será un servicio SOAP síncrono. Entre el proxy y el servicio de negocio tendremos una cola de mensajes punto a punto y un consumidor que sacará los mensajes conforme lleguen y los reenviará al servicio de negocio.

La aplicación de prueba enviará un determinado número de mensajes y todos llegarán al servicio de negocio.

El WSDL generado es: <http://a.b.c.d:8081/Application3-Project1-context-root/WSDLUnidireccionalPort?WSDL>

En este caso tenemos como entrada un string y no nos devuelve nada. Para saber si los mensajes llegan al servicio de negocio (ya que no nos devuelve un resultado) al generar el Web Services, en la clase .java ponemos `System.out.print("Llego al final")` para saber que se ejecuta de manera correcta. Dicho resultado se verá en log del OSB.

Creamos un nuevo proyecto en la consola del OSB llamado ConsumidorDeMensajes.

Creamos tres carpetas: *Proxy Services*, *Business Services* y *Recursos*.



En la carpeta recursos añadimos el wsdl que acabamos de crear:

ConsumidorDeMensajes/Recursos

Referencias

Referencias de otros

Carpeta

Introducir Nombre

Nombre

Suprimir

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Seleccionar Tipo de Recurso

- Servicio
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque
 - Recursos de URL
 - Recursos Comprimidos

Hasta ConsumidorDeMensajes

Agregar Carpeta

Elementos 0 - 0 de 0

No hay ninguna carpeta que mostrar.

Elementos 0 - 0 de 0

Elementos 1 - 2 de 2

Introducimos la URL donde se encuentra el WSDL, añadimos un nombre y seleccionamos el tipo de recurso WSDL. Pulsamos “Siguiente”.

Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

Siguiente >>

Cancelar

Pulsamos “Importar”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 2 de 2

Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
http://.../Application3-Proj...	WSDLUnidireccionalPort	ConsumidorDeMensajes/Recursos	WSDL
http://.../Application3-Proj...	XMLSchema_1901087800	ConsumidorDeMensajes/Recursos	Esquema XML

Elementos 1 - 2 de 2

<< Anterior

Importar

Cancelar

Ya tenemos los recursos necesarios:

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Elementos 1 - 2 de 2

Nombre	Tipo de Recurso	Acciones	Opciones
WSDLUnidireccionalPort	WSDL		añe
XMLSchema_1901087800	Esquema XML		añe

Elementos 1 - 2 de 2

Suprimir

Creamos el *Business Services* a partir del wsdl que hemos creado:

The screenshot shows a web application interface for 'ConsumidorDeMensajes/Business Services'. A dropdown menu is open, listing various resource types under the heading 'Seleccionar Tipo de Recurso'. The categories and their items are:

- Servicio**
 - Servicio de Proxy
 - Servicio de Negocio** (highlighted)
 - División-Unión
- Interfaz**
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación**
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad**
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad**
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque**
 - Recursos de URL
 - Recursos Comprimidos

At the bottom of the dropdown, there is a 'Crear Recurso:' button with a dropdown arrow.

Ponemos un nombre y seleccionamos el wsdl que acabamos de crear en la carpeta recursos. Pulsamos “siguiente”.

The screenshot shows the 'Crear un Servicio de Negocio (ConsumidorDeMensajes/Business Services/)' configuration window. The 'Configuración General' tab is active. The 'Nombre del Servicio*' field contains 'BusinessService1'. The 'Tipo de Servicio*' section is expanded, showing options for creating a new service or from an existing one. The 'Crear un Servicio Nuevo' section is selected, with 'Servicio Web WSDL' chosen. The 'ConsumidorDeMensajes/Recursos/WSDL' path is entered, and the 'Examinar...' button is visible. The 'WSDLUnidireccionalPort' port is also shown. The 'Crear a partir de Servicio Existente' section is not selected. At the bottom, there are buttons for 'Siguiente >>', 'Último >>', and 'Cancelar'.

Seleccionamos el protocolo http (está por defecto) y añadimos una URI de punto final. Dejamos el resto de campos como están y pulsamos “último”.

Crear un Servicio de Negocio (ConsumidorDeMensajes/Business Services/BusinessService2)

Configuración de Transporte

Protocolo*	http
Algoritmo de Equilibrio de Carga	asignación en rueda (round-robin)
URI de Punto Final*	<p>Formato: http://host:port/someService</p> <p>http://localhost:7001/ConsumidorDeMensajes/Business_Services/BusinessSer</p> <p>URI EXISTENTES</p> <p>http:// /Application3-Project1-context-root/WSDLUnidireccionalPort</p>
Recuento de Reintentos	0
Intervalo de Iteración de Reintentos	30
Reintentar tras Errores de Aplicación	<input checked="" type="radio"/> Sí <input type="radio"/> No

<< Anterior Siguiente >> Último >> Cancelar

A continuación creamos el ProxyService2 a partir del BusinessService1

ConsumidorDeMensajes/Proxy Services

Seleccionar Tipo de Recurso

- Servicio**
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz**
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación**
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad**
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad**
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque**
 - Recursos de URL
 - Recursos Comprimidos

Nombre

Suprimir

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Agregar Carpeta

No hay ninguna carpeta que mostrar.

Le damos un nombre: ProxyService2 y lo creamos a partir del servicio de negocio creado en el paso anterior. Pulsamos “siguiente”.

Crear un Servicio de Proxy (ConsumidorDeMensajes/Proxy Services/)

Configuración General

Nombre del Servicio* ProxyService2

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL Examinar...
(puerto o enlace)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☒ Servicio de Negocio Mensajes/Business Services/BusinessService1 Examinar...

☐ Servicio de Proxy Examinar...

Siguiente >> Último >> Cancelar

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola que va a utilizar.

Crear un Servicio de Proxy (ConsumidorDeMensajes/Proxy Services/ProxyService)

Configuración de Transporte

Protocolo* jms

URI de Punto Final* Formato: jms://((host:port)(,((host:port))*|((host:port)?)/FactoryIndName/QueueIndName)
jms://*****:/weblogic.jms.XAConnectionFactory/ProxyServiceReq

Obtener Todas las Cabeceras

☐ Sí

☒ No

Cabecera Agregar

CABECERA	ACCIÓN
No hay ninguna cabecera configurada.	

<< Anterior Siguiente >> Último >> Cancelar

A continuación creamos el Proxy principal al que se accede por HTTP y hace referencia al Proxy anteriormente creado.

Le damos un nombre (ProxyService1) y lo creamos a partir del ProxyService2. Pulsamos "siguiente".

Crear un Servicio de Proxy (ConsumidorDeMensajes/Proxy Services/)

Configuración General

Nombre del Servicio* ProxyService1

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL Examinar...
(puerto o enlace)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio Examinar...

☒ Servicio de Proxy Mensajes/Proxy Services/ProxyService2 Examinar...

Siguiente >> Último >> Cancelar

Cambiamos el protocolo de transporte a HTTP. Pulsamos “siguiente” hasta el final sin hacer ningún cambio más.

Crear un Servicio de Proxy (ConsumidorDeMensajes/Business Services/ProxyService1)

Configuración de Transporte

Protocolo*	http
URI de Punto Final*	Formato: /someName /ConsumidorDeMensajes/Business_Services/ProxyService1
Obtener Todas las Cabeceras	<input checked="" type="radio"/> Sí <input type="radio"/> No

<< Anterior Siguiete >> Último >> Cancelar

A continuación, vamos al flujo de mensajes del ProxyService1.

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Nombre	Tipo de Recurso	Acciones	Opciones
ProxyService1	Servicio de Proxy		a e
ProxyService2	Servicio de Proxy		a e

Elementos 1 - 2 de 2

Suprimir

El OSB ha hecho el direccionamiento directamente al BusinessService1 pero lo que queremos es que pase por la cola JMS. Por ello cambiamos el direccionamiento y elegimos el PS2.

@ **Direccionar a** ProxyService2 llamando a IntroducirNombre

☐ Utilice la operación entrante para la saliente

Acciones de Petición:

Agregar una Acción

Acciones de Respuesta:

Agregar una Acción

Prueba en la consola de prueba de OSB

Ejecutamos el ProxyService1

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Nombre	Tipo de Recurso	Acciones	Opciones
ProxyService1	Servicio de Proxy		a e
ProxyService2	Servicio de Proxy		a e

Elementos 1 - 2 de 2

Suprimir

Pulamos “Ejecutar”. En el Log de la consola aparecerá el mensaje “Llego al final”. De esta manera podemos ver que realiza la llamada al servicio de negocio.

Si entramos en Weblogic en Servidores JMS:



Pulsamos en wlsbJMSServer:

Resumen de Servidores JMS

Los servidores JMS actúan como contenedores de gestión para las colas y los temas en los módulos JMS que están dirigidos a ellos.
En esta página se resumen los servidores JMS que se han creado en el dominio de WebLogic Server actual.

[Personalizar esta Tabla](#)

Servidores JMS (Filtrado: Existen Más Columnas)

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Almacén Persistente	Destino	Servidor Actual	Estado
<input type="checkbox"/>	BPMJMSServer	BPMJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	JMSServer	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	SOAJMSServer	SOAJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	UMSJMServer	UMSJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	wlsbJMSServer	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	WseeJmsServer	WseeFileStore	osb_server1	osb_server1	✓ OK

[Nuevo](#) [Suprimir](#) Mostrando 1 a 6 de 6 Anterior | Siguiente

Vamos a la pestaña Supervisión / Destinos Activos.

Podemos ver como se ha creado la cola del ProxyService2. Si volvemos a ejecutar el ProxyService1 desde la consola del OSB veremos como el número de mensajes recibidos se incrementa en uno.

Valores para wlsbJMServer

Configuración Registro Destinos **Supervisión** Control Notas

Supervisión **Destinos Activos** Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

Mostrando 1 a 10 de 14 Anterior Siguiente

	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!aaRequest-919366798	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ooRequest-1695944011	1	0	1	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ProxyService2Request-1304076865	0	0	4	13	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request899767483	0	0	1	7	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request-1184180074	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	2	21	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción Consumo Inserción

Mostrando 1 a 10 de 14 Anterior Siguiente

A continuación paramos la cola, consumo / parar:

Valores para wlsbJMServer

Configuración Registro Destinos **Supervisión** Control Notas

Supervisión **Destinos Activos** Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

Mostrando 1 a 10 de 14 Anterior Siguiente

	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!aaRequest-919366798	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ooRequest-1695944011	1	0	1	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input checked="" type="checkbox"/>	jmsResources!ProxyService2Request-1304076865	0	0	4	13	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request899767483	0	0	1	7	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request-1184180074	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	2	21	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción Consumo Inserción

Mostrando 1 a 10 de 14 Anterior Siguiente

Si volvemos a ejecutar ahora el ProxyService1 desde la consola del OSB unas cuantas veces, veremos como el número de mensajes acumulados en la cola va aumentando así como el número de mensajes recibidos pero por el Log de la consola no se muestra el mensaje "Llego al final" ya que la cola no está enviando los mensajes al servicio de negocio.

Valores para wlsbjmsServer											
Configuración Registro Destinos Supervisión Control Notas											
Supervisión Destinos Activos Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas											
Esta página permite ver destinos activos dirigidos a este servidor JMS.											
Personalizar esta Tabla											
Destinos (Filtrado: Existen Más Columnas)											
Producción Consumo Inserción Mostrando 1 a 10 de 14 Anterior Siguiente											
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!aaRequest-919366798	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ooRequest-1695944011	1	0	1	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ProxyService2Request-1304076865	3	0	4	16	0	Queue	paused_consumption, advertised_in_cluster_jndi	false	false	true
<input type="checkbox"/>	jmsResources!PS22Request899767483	0	0	1	7	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request-1184180074	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	2	21	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
Producción Consumo Inserción Mostrando 1 a 10 de 14 Anterior Siguiente											

Si volvemos a reanudar el consumo, vemos como el número de mensajes actuales pasa a 0 y en el Log de la consola aparecerá 3 veces (en este caso) el mensaje “Llego al final”.

Valores para wlsbjmsServer											
Configuración Registro Destinos Supervisión Control Notas											
Supervisión Destinos Activos Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas											
Esta página permite ver destinos activos dirigidos a este servidor JMS.											
Personalizar esta Tabla											
Destinos (Filtrado: Existen Más Columnas)											
Producción Consumo Inserción Mostrando 1 a 10 de 14 Anterior Siguiente											
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!aaRequest-919366798	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ooRequest-1695944011	1	0	1	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ProxyService2Request-1304076865	0	3	4	16	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS22Request899767483	0	0	1	7	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request-1184180074	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	2	21	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
Producción Consumo Inserción Mostrando 1 a 10 de 14 Anterior Siguiente											

Actualizamos la pantalla:

Valores para wlsbJMSServer

Configuración

Registro

Destinos

Supervisión

Control

Notas

Supervisión

Destinos Activos

Transacciones Activas

Conexiones Activas

Pools de Sesiones Activas

Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Producción

Consumo

Inserción

Mostrando 1 a 10 de 14 · Anterior · Siguiente

<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!aaRequest-919366798	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ooRequest-1695944011	1	0	1	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ProxyService2Request-1304076865	0	0	4	16	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request899767483	0	0	1	7	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request-1184180074	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	2	21	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción

Consumo

Inserción

Mostrando 1 a 10 de 14 · Anterior · Siguiente

Caso de Uso 2: Canal punto a punto confiable

Paso 1

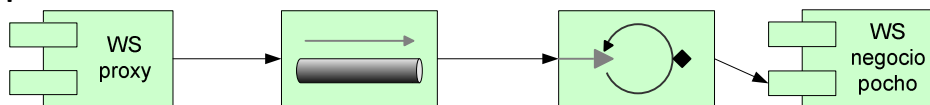
Tendremos un servicio Web SOAP de negocio de un solo sentido que fallará algunas veces, devolviendo una excepción.

Una aplicación cliente de prueba llamará al servicio un número determinado de veces. Sólo algunas de las llamadas tendrán éxito, a causa de los fallos del servicio.

Paso 2

Se desplegará el servicio de negocio en el BUS y se publicará un servicio proxy.

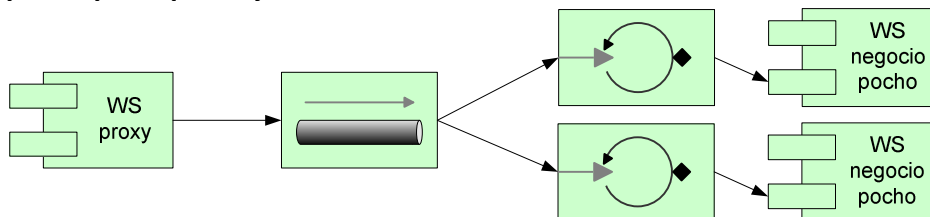
Entre el proxy y el servicio de negocio se implementará un canal de mensajes punto a punto.



La aplicación cliente llamará al proxy, y, a pesar de los fallos del servicio, todas las llamadas tendrán éxito y el servicio de negocio recibirá todos los mensajes.

Paso 3

Se desplegará otra instancia del servicio de negocio y se configurará el canal punto a punto para que haya varios consumidores.



La aplicación cliente llamará al proxy, y, a pesar de los fallos de los servicios, todas las llamadas tendrán éxito, todos los mensajes habrán llegado a una u otra instancia del servicio de negocio y ningún mensaje se habrá recibido dos veces.

PASO 1

URL wsdlPocho:

<http://a.b.c.d:8081/Application4-WSDLUnidireccionalPocho-context-root/WSDLPochoPort?WSDL>

Se ha desarrollado un *Web Service* que falla a veces. Si el segundo en el que se llama a dicho *Web Service* es mayor que 30 funciona correctamente, en otro caso lanza una excepción. De esta manera tenemos un servicio que no siempre funciona. Prueba desde el cliente:

Si el segundo es mayor que 30:

IntroducirNombre Request Summary	
Arguments:	string arg0:
Returned:	[void]
Submitted:	Fri Dec 17 11:08:33 CET 2010
Duration:	1 ms
Re-Invoke	

IntroducirNombre Request Detail	
Service Request	
<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header /> <env:Body> <IntroducirNombre xmlns="http://wsdlunidireccionalpocho/"> <!--Optional--> <arg0 xmlns="" /> </IntroducirNombre> </env:Body> </env:Envelope></pre>	
Service Response	
<pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <ns2:IntroducirNombreResponse xmlns:ns2="http://wsdlunidireccionalpocho/" /> </S:Body> </S:Envelope></pre>	

Sino:

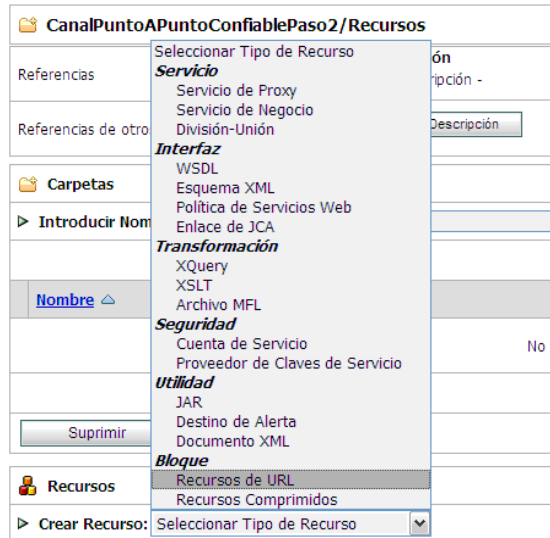
IntroducirNombre Request Summary	
Arguments:	string arg0:
Fault:	Ha ocurrido un error
Submitted:	Fri Dec 17 11:06:21 CET 2010
Duration:	7 ms
Re-Invoke	

IntroducirNombre Request Detail	
Service Request	
<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header /> <env:Body> <IntroducirNombre xmlns="http://wsdlunidireccionalpocho/"> <!--Optional--> <arg0 xmlns="" /> </IntroducirNombre> </env:Body> </env:Envelope></pre>	
Service Response	
<pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope"> <faultcode>S:Server</faultcode> <faultstring>Ha ocurrido un error</faultstring> <detail> <ns2:Exception xmlns:ns2="http://wsdlunidireccionalpocho/"> <message>Ha ocurrido un error</message> </ns2:Exception> <ns2:exception xmlns:ns2="http://jax-ws.dev.java.net/" class="java.lang.Exception" note="To disable this feature, set com.sun.xml.ws.fault.SOAPFaultBuilder.disableCaptureStackTrace system property to false"> <message>Ha ocurrido un error</message> <ns2:stackTrace> <ns2:frame class="wsdlunidireccionalpocho.WSDLPocho" file="WSDLPocho.java" line="26" method="IntroducirNombre" /> <ns2:frame class="sun.reflect.GeneratedMethodAccessor1000" line="unknown" method="invoke" /> <ns2:frame class="sun.reflect.DelegatingMethodAccessorImpl" file="DelegatingMethodAccessorImpl.java" line="25" method="invoke" /> <ns2:frame class="java.lang.reflect.Method" file="Method.java" line="597" method="invoke" /> <ns2:frame class="weblogic.wsee.jaxws.WLSInstanceResolver\$WLSInvoker" file="WLSInstanceResolver.java" line="92" method="invoke" /> <ns2:frame class="weblogic.wsee.jaxws.WLSInstanceResolver\$WLSInvoker" file="WLSInstanceResolver.java" line="74" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.InvokerTube\$2" file="InvokerTube.java" line="151" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.sei.EndpointMethodHandlerImpl" file="EndpointMethodHandlerImpl.java" line="265" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.sei.SEIInvokerTube" file="SEIInvokerTube.java" line="100" method="processRequest" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="604" method="__doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="563" method="__doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="548" method="doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="445" method="runSync" /> <ns2:frame class="com.sun.xml.ws.server.WSEndpointImpl\$2" file="WSEndpointImpl.java" line="373" method="process" /> <ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter\$HttpToolkit" file="HttpAdapter.java" line="524" method="handle" /> <ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter" file="HttpAdapter.java" line="255" method="handle" /> <ns2:frame class="com.sun.xml.ws.transport.http.servlet.ServletAdapter" file="ServletAdapter.java" line="140" method="handle" /> <ns2:frame class="weblogic.wsee.jaxws.WLSServletAdapter" file="WLSServletAdapter.java" line="208" method="handle" /> <ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter\$AuthorizedInvoke" file="HttpServletAdapter.java" line="310" method="run" /> <ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter" file="HttpServletAdapter.java" line="223" method="post" /> <ns2:frame class="weblogic.wsee.jaxws.JAXWServlet" file="JAXWServlet.java" line="124" method="doPost" /> <ns2:frame class="javax.servlet.http.HttpServlet" file="HttpServlet.java" line="727" method="service" /> </ns2:stackTrace> </ns2:exception> </detail> </S:Fault> </S:Body> </S:Envelope></pre>	

PASO 2

Creamos un nuevo proyecto llamado CanalPuntoAPuntoConfiablePaso2. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

En la carpeta Recursos añadimos el wsdlPocho.



Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

“Siguiente”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 2 de 2

<input checked="" type="checkbox"/>	Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	http://.../Application4-WSDL...	WSDLPochoPort	CanalPuntoAPuntoConfiablePaso2/Recursos	WSDL
<input checked="" type="checkbox"/>	http://.../Application4-WSDL...	XMLSchema_1159110265	CanalPuntoAPuntoConfiablePaso2/Recursos	Esquema XML

Elementos 1 - 2 de 2

“Importar”.

Creamos en la carpeta *Business Services* un servicio de negocio a partir del wsdl que hemos creado:

CanalPuntoAPuntoConfiablePaso2/Business Services

Referencias	Seleccionar Tipo de Recurso Servicio Servicio de Proxy Servicio de Negocio División-Unión	Descripción No hay ninguna carpeta que mostrar.
Referencias de otro	Interfaz WSDL Esquema XML Política de Servicios Web Enlace de JCA	
Carpetas	Transformación XQuery XSLT Archivo MFL	
Introducir Nombre	Seguridad Cuenta de Servicio Proveedor de Claves de Servicio	
Nombre	Utilidad JAR Destino de Alerta Documento XML	
Suprimir	Bloque Recursos de URL Recursos Comprimidos	
Recursos	Seleccionar Tipo de Recurso	

Crear Recurso: Seleccionar Tipo de Recurso

Ponemos un nombre y seleccionamos el wsdl que acabamos de crear en la carpeta recursos. “Siguiente”.

Crear un Servicio de Negocio (CanalPuntoAPuntoConfiablePaso2/Business Services/)

Configuración General

Nombre del Servicio*: BusinessService1

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

CanalPuntoAPuntoConfiablePaso2/Recursos/WSDLPochoPort (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> Último >> Cancelar

Seleccionamos el protocolo http (está por defecto) y añadimos una URI de punto final. Dejamos el resto de campos como están y pulsamos “último”.

Crear un Servicio de Negocio (CanalPuntoAPuntoConfiablePaso2/Business Services/BusinessService11)

Configuración de Transporte

Protocolo*: http

Algoritmo de Equilibrio de Carga: asignación en rueda (round-robin)

URI de Punto Final*

Formato: http://host:port/someService

http://localhost:7001/CanalPuntoAPuntoConfiablePaso2/Business_Services/B

URI EXISTENTES

http://.../Application4-WSDLUnidireccionalPocho-context-root/WSDLPochoPort

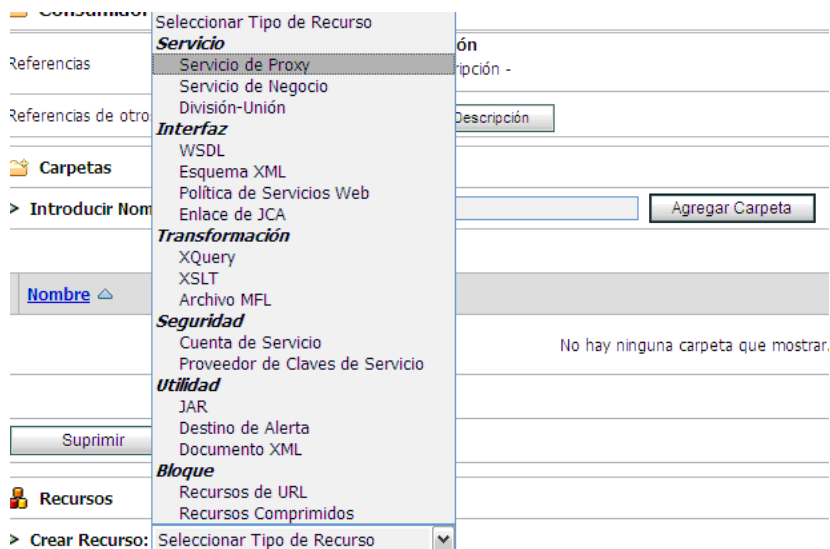
Recuento de Reintentos: 0

Intervalo de Iteración de Reintentos: 30

Reintentar tras Errores de Aplicación: ☒ Sí ☐ No

<< Anterior Siguiente >> Último >> Cancelar

A continuación creamos el PS2 a partir del BusinessService1



Le damos un nombre: PS2 y lo creamos a partir de cualquier servicio SOAP. "Siguiente".

Crear un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/)

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración (puerto o enlace)

☐ Servicio de Mensajes

☒ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola que va a utilizar. Las colas se crean de manera automática una vez que ejecutemos el Proxy.

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)

Configuración de Transporte

Protocolo*

URI de Punto Final*

Obtener Todas las Cabeceras ☒ Sí ☐ No

"Siguiente".

Seleccionamos la siguiente configuración de transporte:

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)

Configuración de Transporte de JMS

Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema
Se Necesita Respuesta	<input checked="" type="checkbox"/>
Patrón de Respuesta	<input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID
Tipo de Mensaje de Respuesta	<input type="radio"/> Bytes <input checked="" type="radio"/> Texto
Política de Distribución	default
Codificación de Peticiones	UTF-8
Codificación de Respuestas	UTF-8
Timeout de Respuesta de Cliente	300
URI de Respuesta	jdbc:ims.XAConnectionFactory/Response
Cuenta de Servicio de JMS	<input type="text"/> Examinar...

Configuración Avanzada

<< Anterior Siguiente >> Último >> Cancelar

Desplegamos Configuración avanzada.

Introducimos el número de reintentos en caso de fallo, el tiempo entre cada uno de esos reintentos (en milisegundos), el destino de error (en este caso la cola ColaError) y en caso de que después de todos los reintentos siga fallando se redireccione a esta cola.

Configuración Avanzada

Usar SSL	<input type="checkbox"/>
Selector de Mensajes	<input type="text"/>
Suscripción Duradera	<input type="checkbox"/>
Recuento de Reintentos	2
Intervalo de Reintentos	5000
Destino de Error	ColaError
Política de Caducidad	Redireccionar
Se necesita XA	<input type="checkbox"/>
Timeout de JNDI	0

<< Anterior Siguiente >> Último >> Cancelar

“Siguiente”.

“Siguiente”. Seleccionamos misma transacción para respuesta. Si algo ocurre se produce rollback en la transacción.

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)

Manejo de Mensajes


Transacción Necesaria	<input type="checkbox"/> Activado
Misma Transacción para Respuesta	<input checked="" type="checkbox"/> Activado
Flujo de Contenido	<input type="checkbox"/> Activado Tipo de Buffer <input type="radio"/> Buffer de Memoria <input type="radio"/> Buffer de Disco Compresión <input type="checkbox"/> Activado

<< Anterior Siguiente >> Último >> Cancelar

“Último” y “Guardar”.


Entramos en el flujo de mensajes del PS2 que acabamos de crear. Añadimos “Direccionamiento” al BusinessService1. Añadimos “Opciones de direccionamiento”.

Seleccionamos Calidad del Servicio “Una Sola Vez”, para evitar que se dupliquen los mensajes.

 @ **Direccionar a BusinessService1*** llamando a **IntroducirNombre** ▼


☐ Utilice la operación entrante para la saliente

Acciones de Petición:

 @ **Opciones de Direccionamiento** para sustituir las siguientes configuraciones por defecto:


☐ URI: <Expresión>*
☒ Calidad del Servicio: **Una Sola Vez** ▼*
☐ Modo: **Petición** ▼*
☐ Intervalo de Reintentos: * (segundo)
☐ Recuento de Reintentos: *
☐ Prioridad: <Expresión>*

Acciones de Respuesta:

 [Agregar una Acción](#)

A continuación creamos el Proxy principal al que se accede por HTTP y hace referencia al Proxy anteriormente creado.

Le damos un nombre (PS1) y lo creamos a partir del WSDL. “Siguiente”.

 **Crear un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/)**

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo


☒ Servicio Web WSDL
 (puerto)

☐ Servicio de Transporte según Configuración
☐ Servicio de Mensajes
☐ Cualquier Servicio SOAP
☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio
☐ Servicio de Proxy

Dejamos el protocolo de transporte a HTTP. Pulsamos “siguiente” hasta el final sin hacer ningún cambio más.

 Crear un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS11)

Configuración de Transporte

Protocolo*	http
URI de Punto Final*	Formato: /someName /CanalPuntoAPuntoConfiablePaso2/Proxy_Services/PS1
Obtener Todas las Cabeceras	<input checked="" type="radio"/> Sí <input type="radio"/> No

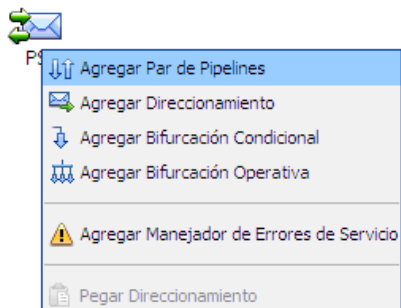
<< Anterior Siguiente >> Último >> Cancelar

Por último vamos al flujo de mensajes del PS1.

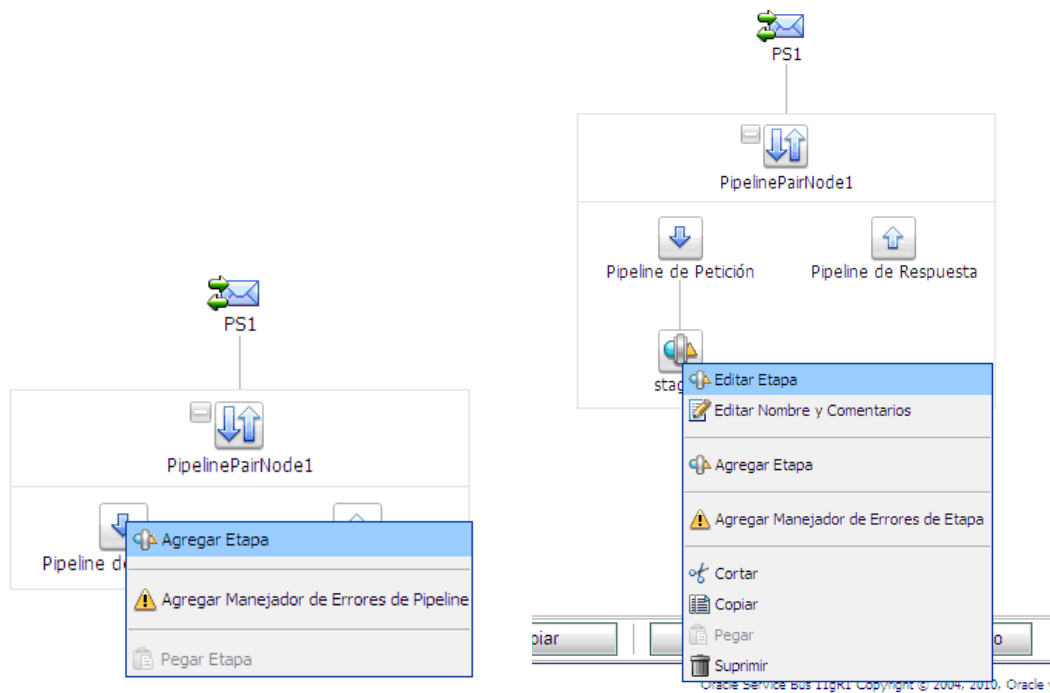
Recursos			
▶ Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 2 de 2			
<input type="checkbox"/> Nombre	Tipo de Recurso	Acciones	Opciones
<input type="checkbox"/> PS1	Servicio de Proxy		aTe 
<input type="checkbox"/> PS2	Servicio de Proxy		aTe 
Elementos 1 - 2 de 2			
<input type="button" value="Suprimir"/>			

Como queremos mandar el mensaje pero no queremos esperar a la respuesta sino obtener una respuesta inmediata debemos utilizar “Publish” en vez de “Route to”. Por ello:

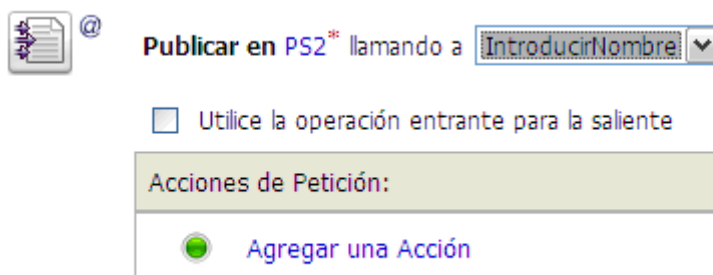
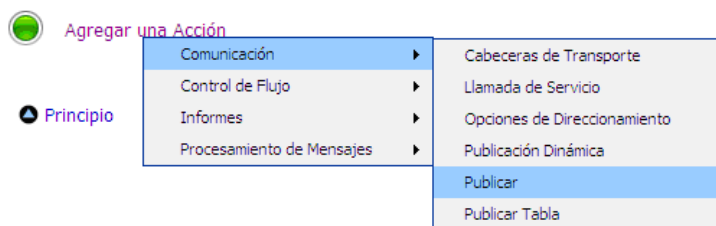
- Añadimos un par de pipelines



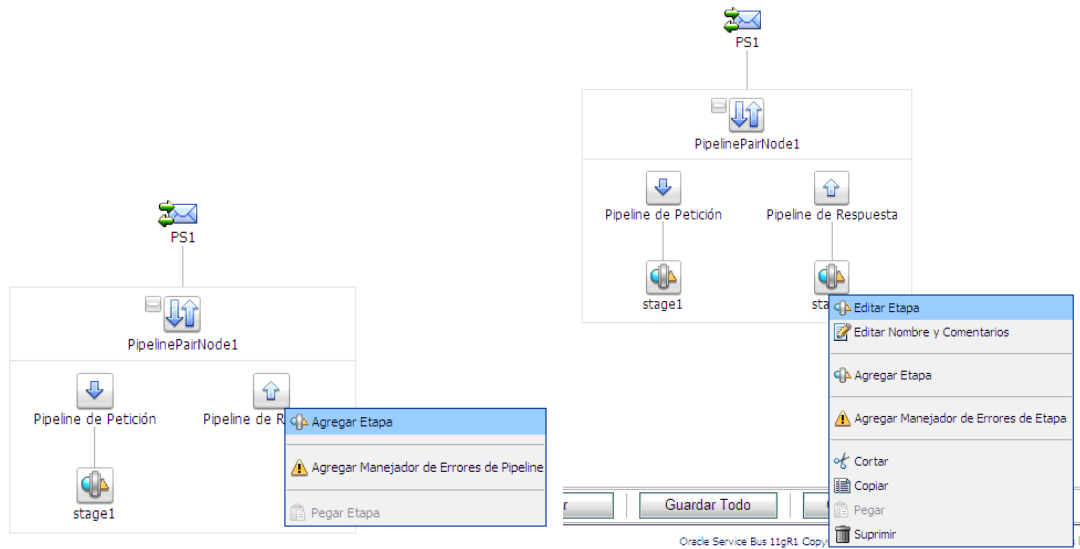
- En el pipeline de petición añadimos una nueva etapa.



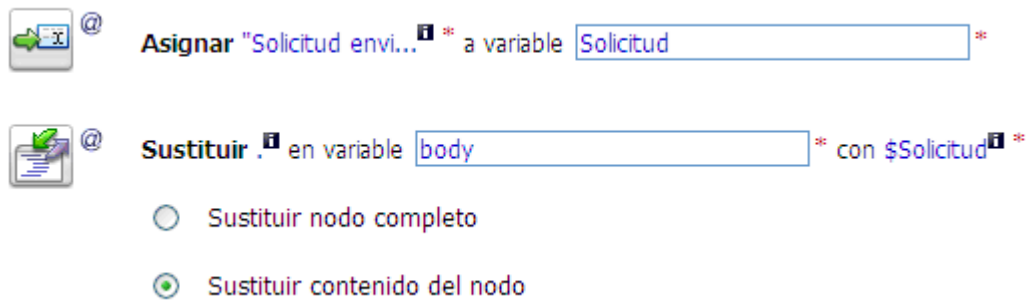
6. Añadimos una nueva acción “Publicar”. Seleccionamos al servicio al que queremos direccionar, en este caso PS2 y seleccionamos el único método que existe.



7. En el pipeline de respuesta añadimos una nueva etapa.



8. Añadimos una acción “Asignar” para crear la respuesta del Proxy.



Prueba en la consola del OSB

Ejecutamos varias veces el PS1.

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Elementos 1 - 2 de 2

1

	Nombre	Tipo de Recurso	Acciones	Opciones
	PS1	Servicio de Proxy	<div><div><div></div></div><div><div></div></div><div><div></div></div></div>	<div><div></div></div> <div><div></div></div> <div><div></div></div>

| | PS2 | Servicio de Proxy | | |

Elementos 1 - 2 de 2

1

Suprimir

Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación.

Prueba de Servicios de Proxy - PS1 Ayuda

Atrás Cerrar

Documento de Petición

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <wsd:IntroducirNombre xmlns:wsd="http://wsdlunidireccionalpocho/">
      <!--Optional:-->
      <arg0>string</arg0>
    </wsd:IntroducirNombre>
  </soapenv:Body>
</soapenv:Envelope>
```

Documento de Respuesta

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>Solicitud enviada</soapenv:Body>
</soapenv:Envelope>
```

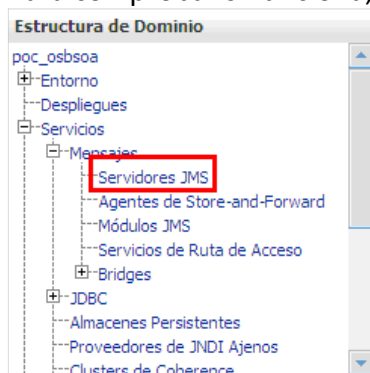
Metadatos de Respuesta

```
<con:metadata xmlns:con="http://www.bea.com/wli/sb/test/config">
  <tran:headers xsi:type="http:HttpHeaders" xmlns:http="http://www.bea.com/wli/sb/transport/http"
    xmlns:tran="http://www.bea.com/wli/sb/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <http:Content-Type>text/xml; charset=utf-8</http:Content-Type>
  </tran:headers>
  <tran:response-code xmlns:tran="http://www.bea.com/wli/sb/transport">0</tran:response-code>
  <tran:encoding xmlns:tran="http://www.bea.com/wli/sb/transport">utf-8</tran:encoding>
</con:metadata>
```

Rastreo de Llamada

- (recibiendo petición)
- PipelinePairNode1
 - stage1
- (realizando eco de petición)
- PipelinePairNode1
 - stage1

Para comprobar si funciona, vamos al weblogic. Servicios / Mensajes / Servidores JMS



Elegimos el servidor wlsbJMSServer.

Resumen de Servidores JMS

Los servidores JMS actúan como contenedores de gestión para las colas y los temas en los módulos JMS que están dirigidos a ellos.

En esta página se resumen los servidores JMS que se han creado en el dominio de WebLogic Server actual.

[Personalizar esta Tabla](#)

Servidores JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior Siguiente

<input type="checkbox"/>	Nombre	Almacén Persistente	Destino	Servidor Actual	Estado
<input type="checkbox"/>	BPMJMServer	BPMJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	JMSserver	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	SOAJMServer	SOAJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	UMSJMServer	UMSJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	wsbJMServer	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	WseeJmsServer	WseeFileStore	osb_server1	osb_server1	✓ OK

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior Siguiente

Pestaña Supervisión / Destinos Activos.

En este ejemplo, se ha ejecutado el PS1 6 veces, de las cuales 4 han tenido éxito y 2 han fallado.

Valores para wsbJMServer

Configuración Registro Destinos **Supervisión** Control Notas

Supervisión **Destinos Activos** Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

[Personalizar esta Tabla](#)

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

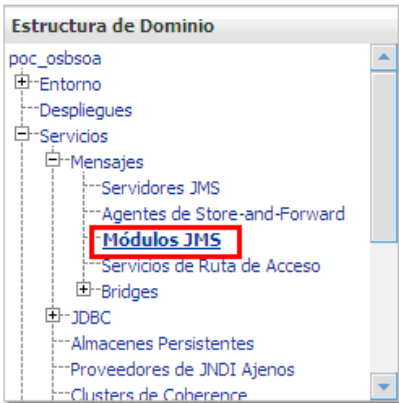
Mostrando 1 a 10 de 12 Anterior Siguiente

<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!ColaError	2	0	2	2	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2Request1168862377	0	0	0	0	0	Queue	suspended	false	false	false
<input type="checkbox"/>	jmsResources!QueueIn	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!Request-1972615808	0	0	2	6	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!Response552119583	0	0	1	4	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción Consumo Inserción

Mostrando 1 a 10 de 12 Anterior Siguiente

Si entramos en Servicios / Módulos JMS, podemos ver los mensajes que se han guardado en la cola de errores.



Módulo jmsResources.

Módulos JMS

Los recursos del sistema de JMS se configuran y almacenan como módulos parecidos a los módulos de J2EE estándar. Dichos recursos incluyen colas, temas, fábricas de conexiones, plantillas, claves de destino, cuotas, colas distribuidas, temas distribuidos, servidores ajenos y parámetros de Store-and-Forward (SAF) de JMS. Puede configurar y gestionar de forma administrativa módulos de sistema de JMS como recursos del sistema globales.

En esta página se resumen los módulos del sistema de JMS creados para este dominio.

[Personalizar esta Tabla](#)

Módulos JMS

Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/> Nombre	Tipo
<input type="checkbox"/> BPMJMSModule	Sistema
<input type="checkbox"/> jmsResources	Sistema
<input type="checkbox"/> SOAJMSModule	Sistema
<input type="checkbox"/> SystemModule	Sistema
<input type="checkbox"/> UMSJMSSystemResource	Sistema
<input type="checkbox"/> WseeJmsModule	Sistema

Mostrando 1 a 6 de 6 Anterior | Siguiente

Nombre de la cola: ColaError.

[Personalizar esta Tabla](#)

Resumen de Recursos

Mostrando 1 a 10 de 15 Anterior | Siguiente

<input type="checkbox"/> Nombre	Tipo	Nombre de JNDI	Despliegue Secundario	Destinos
<input type="checkbox"/> ColaError	Cola	ColaError	ColaError	wlsbjmsServer
<input type="checkbox"/> QueueIn	Cola	QueueIn	wlsbjmsServer	wlsbjmsServer
<input type="checkbox"/> QueueIn.Quota	Cuota	N/A	N/A	N/A
<input type="checkbox"/> Request-1972615808	Cola	Request	Request-1972615808	wlsbjmsServer
<input type="checkbox"/> Response552119583	Cola	Response	Response552119583	wlsbjmsServer
<input type="checkbox"/> TemporaryTplmt	Plantilla	N/A	N/A	N/A
<input type="checkbox"/> weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Fábrica de Conexiones	weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/> wli.reporting.jmsprovider.ConnectionFactory	Fábrica de Conexiones	wli.reporting.jmsprovider.ConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/> wli.reporting.jmsprovider.queue	Cola	wli.reporting.jmsprovider.queue	wlsbjmsServer	wlsbjmsServer
<input type="checkbox"/> wli.reporting.jmsprovider_error.queue	Cola	wli.reporting.jmsprovider_error.queue	wlsbjmsServer	wlsbjmsServer

Mostrando 1 a 10 de 15 Anterior | Siguiente

Pestaña Supervisión.

Seleccionamos la cola y pulsamos “Mostrar Mensajes”.

Valores para ColaError

Configuración **Supervisión** Control Seguridad Despliegue Secundario Notas

Un destino de JMS identifica una cola (punto a punto) o un tema (publicación/suscripción) dirigidos a un servidor JMS.

En esta página se resumen los destinos de JMS activos que se han creado para este módulo JMS.

[Personalizar esta Tabla](#)

Destinos (Filtrado: Existen Más Columnas)

[Mostrar Mensajes](#) Mostrando 1 a 1 de 1 Anterior | Siguiente

<input checked="" type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Mensajes Totales	Consumidores Actuales	Recuento Alto de Consumidores	Consumidores Totales	Recuento Alto de Mensajes
<input checked="" type="checkbox"/>	jmsResource/ColaError	2	0	2	0	0	0	2

[Mostrar Mensajes](#) Mostrando 1 a 1 de 1 Anterior | Siguiente

En este caso se han producido dos mensajes erróneos, en el segundo 13 y 16. Se puede ver el contenido de dichos mensajes.

Resumen de Mensajes de JMS

En esta página se resumen los mensajes disponibles para una cola autónoma, una cola distribuida o un suscriptor duradero de temas. Utilice esta página para ver detalles de mensajes, crear nuevos mensajes, suprimir mensajes seleccionados, mover mensajes a otro destino, exportar contenidos de mensajes en formato XML a otro archivo, importar contenido de mensajes con formato XML de otro archivo o vaciar todos los mensajes de un destino.

Haga clic en un mensaje para ver su contenido.

Selector de Mensajes: [Aplicar](#)

[Personalizar esta Tabla](#)

Mensajes de JMS (Filtrado: Existen Más Columnas)

[Nuevo](#) [Suprimir](#) [Mover](#) [Importar](#) [Exportar](#) Mostrando 1 a 2 de 2 Anterior | Siguiente

<input type="checkbox"/>	Identificador	Identificador de Correlación	Registro de Hora	Cadena de Estado	Modo de Entrega de JMS	Tamaño del Mensaje
<input type="checkbox"/>	ID: <549585.1292593453730.0>	ID:42454153112d310d9f8f23770000012cea06e6f7ffff87f4	Fri Dec 17 CET 2010 14:44:13	visible	Persistente	432
<input type="checkbox"/>	ID: <549585.1292593756071.0>	ID:42454153112d310d9f8f23770000012cea06e6f7ffff8800	Fri Dec 17 CET 2010 14:49:16	visible	Persistente	432

[Nuevo](#) [Suprimir](#) [Mover](#) [Importar](#) [Exportar](#) Mostrando 1 a 2 de 2 Anterior | Siguiente

PASO 3 (ver caso de uso Escalabilidad en canal punto a punto)

Prueba

URL EscalabilidadServiciosSincronosPaso3 OSB:

http://a.b.c.d:8081/CanalPuntoAPuntoConfiablePaso3/Proxy_Services/WSPocho_PS1?wsdl

Ejecutamos varias veces el PS1.

Recursos

► Crear Recurso:

Elementos 1 - 2 de 2

Nombre	Tipo de Recurso	Acciones	Opciones
PS1	Servicio de Proxy		 
PS2	Servicio de Proxy		 

Elementos 1 - 2 de 2

Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación.

Prueba de Servicios de Proxy - PS1 Ayuda

Documento de Petición

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <wsd:IntroducirNombre xmlns:wsd="http://wsdlunidireccionalpocho/">
      <!--Optional:-->
      <arg0>string</arg0>
    </wsd:IntroducirNombre>
  </soapenv:Body>
</soapenv:Envelope>
```

Documento de Respuesta

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>Solicitud enviada</soapenv:Body>
</soapenv:Envelope>
```

Metadatos de Respuesta

```
<con:metadata xmlns:con="http://www.bea.com/wli/sb/test/config">
  <tran:headers xsi:type="http:HttpResponseHeaders" xmlns:http="http://www.bea.com/wli/sb/transport/http"
    xmlns:tran="http://www.bea.com/wli/sb/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <http:Content-Type>text/xml; charset=utf-8</http:Content-Type>
  </tran:headers>
  <tran:response-code xmlns:tran="http://www.bea.com/wli/sb/transport">0</tran:response-code>
  <tran:encoding xmlns:tran="http://www.bea.com/wli/sb/transport">utf-8</tran:encoding>
</con:metadata>
```

Rastreo de Llamada

- (recibiendo petición)
- PipelinePairNode1
 - stage1
- (realizando eco de petición)
- PipelinePairNode1
 - stage1

En la consola del OSB se puede llevar un seguimiento de las llamadas.

Accedemos a Operaciones / Estado de servicio.

Seleccionamos el servicio de negocio creado WSPocho_BS.

ORACLE® Service Bus 11gR1

Centro de Cambios Bienvenido, weblogic Conectado a: poc_osbsoa Inicio Consola de Oracle W

Sesión de web

Alertas de SLA Alertas de Pipeline **Estado de Servicio** Estado del Servidor 1

Estado de Servicio

Mostrar Estadísticas Intervalo de Agregación Actual

Servidor osb_server1

Búsqueda

Nombre

Ruta de Acceso

Filtros de Estado de Servicio

Búsqueda Ver Todo

Nombre	Ruta de Acceso	Tipo de Servicio	Intervalo de Agregación	Tiempo Medio de Respuesta	Men
PS2	CanalPuntoAPuntoConfiablePaso3/Proxy Ser...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio de Seguimiento	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio de Seguimiento (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes	ServicioDeclaracionTelematica/SPRINT 1/B...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	
Servicio Gestor de Soportes (proxy)	ServicioDeclaracionTelematica/SPRINT 1/P...	Servicio de Proxy	0 hora(s) 1 minuto(s)	0 msecs.	
WSPocho_BS	CanalPuntoAPuntoConfiablePaso3/Business ...	Servicio de Negocio	0 hora(s) 1 minuto(s)	0 msecs.	

Principio

Operaciones

Explorador de Recursos

Explorador de Proyectos

Configuración de Seguridad

Administración del Sistema

Pestaña URI de Punto Final. Comprobamos el reparto de los mensajes.

Métricas de Servicio URI de Punto Final

Elementos 1 - 2 de 2

URI de Punto Final	Recuento de Mensajes	Recuento de Errores	Tiempo Mínimo de Respuesta	Tiempo Máximo de Respuesta	Tiempo Medio de Respuesta
http://10.238.7.6:8011/Application4-WSDLUnidireccionalP...	1	0	15 msecs.	15 msecs.	15 msecs.
http://10.238.7.6:8011/Application55-Project1-context-r...	2	0	3 msecs.	15 msecs.	9 msecs.

Elementos 1 - 2 de 2

Atrás Restablecer Estadísticas Refrescar

Si paramos los dos *Web Services* desplegados en *Weblogic* y lanzamos la aplicación, la petición (el mensaje) se guarda en la cola de errores JMS.

Si paramos un único *Web Service*, todas las peticiones serán tratadas por el *Web Service* que queda activo.

Caso de Uso 3: Canal de publicación / subscripción

Alcance:

El objetivo es demostrar la implementación de la integración basada en mensajes con un canal publicador/subscriptor y su tolerancia a fallos.

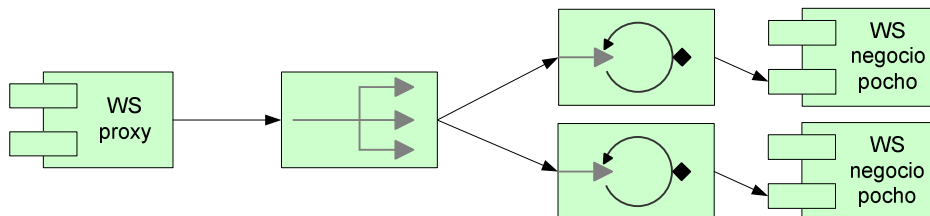
Paso 1

Tendremos dos servicios Web SOAP de negocio de un solo sentido que fallarán algunas veces, devolviendo una excepción. Pueden ser dos instancias del mismo servicio.

Una aplicación cliente de prueba llamará a los dos servicios un número determinado de veces. Sólo algunas de las llamadas tendrán éxito, a causa de los fallos de los servicios.

Paso 2

Se desplegarán los servicios de negocio en el BUS y se publicará un servicio proxy. Entre el proxy y los servicios de negocio se implementará un canal de mensajes publicador/subscriptor.



La aplicación cliente llamará al proxy para enviar una sólo vez cada mensaje y, a pesar de los fallos de los servicios, todas las llamadas tendrán éxito y cada uno de los servicios de negocio recibirá una copia de cada uno de los mensajes.

PASO 1

URL wsdlPocho:

<http://a.b.c.d:8081/Application4-WSDLUnidireccionalPocho-context-root/WSDLPochoPort?WSDL>

Se ha desarrollado un *Web Service* que falla a veces. Si el segundo en el que se llama a dicho *Web Service* es mayor que 30 funciona correctamente, en otro caso lanza una excepción. De esta manera tenemos un servicio que no siempre funciona. Prueba desde el cliente:

Si el segundo es mayor que 30:

IntroducirNombre Request Summary	
Arguments:	string arg0:
Returned:	[void]
Submitted:	Fri Dec 17 11:08:33 CET 2010
Duration:	1 ms
Re-Invoke	

IntroducirNombre Request Detail	
Service Request	
<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header /> <env:Body> <IntroducirNombre xmlns="http://wsdlunidireccionalpocho/"> <!--Optional:--> <arg0 xmlns="" /> </IntroducirNombre> </env:Body> </env:Envelope></pre>	
Service Response	
<pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <ns2:IntroducirNombreResponse xmlns:ns2="http://wsdlunidireccionalpocho/" /> </S:Body> </S:Envelope></pre>	

Sino:

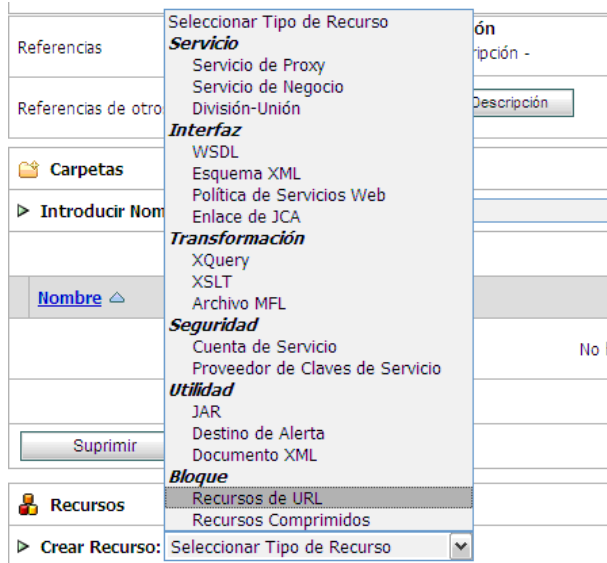
IntroducirNombre Request Summary	
Arguments:	string arg0:
Fault:	Ha ocurrido un error
Submitted:	Fri Dec 17 11:06:21 CET 2010
Duration:	7 ms
Re-Invoke	

IntroducirNombre Request Detail	
Service Request	
<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header /> <env:Body> <IntroducirNombre xmlns="http://wsdlunidireccionalpocho/"> <!--Optional:--> <arg0 xmlns="" /> </IntroducirNombre> </env:Body> </env:Envelope></pre>	
Service Response	
<pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope"> <faultcode>S:Server</faultcode> <faultstring>Ha ocurrido un error</faultstring> <detail> <ns2:Exception xmlns:ns2="http://wsdlunidireccionalpocho/"> <message>Ha ocurrido un error</message> </ns2:Exception> <ns2:exception xmlns:ns2="http://jax-ws.dev.java.net/" class="java.lang.Exception" note="To disable this feature, set com.sun.xml.ws.fault.SOAPFaultBuilder.disableCaptureStackTrace system property to false"> <message>Ha ocurrido un error</message> <ns2:stackTrace> <ns2:frame class="wsdlunidireccionalpocho.WSDLPocho" file="WSDLPocho.java" line="26" method="IntroducirNombre" /> <ns2:frame class="sun.reflect.GeneratedMethodAccessor1000" line="unknown" method="invoke" /> <ns2:frame class="sun.reflect.DelegatingMethodAccessorImpl" file="DelegatingMethodAccessorImpl.java" line="25" method="invoke" /> <ns2:frame class="java.lang.reflect.Method" file="Method.java" line="597" method="invoke" /> <ns2:frame class="weblogic.wsee.jaxws.WLSInstanceResolver\$WLSInvoker" file="WLSInstanceResolver.java" line="92" method="invoke" /> <ns2:frame class="weblogic.wsee.jaxws.WLSInstanceResolver\$WLSInvoker" file="WLSInstanceResolver.java" line="74" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.InvokerTube\$2" file="InvokerTube.java" line="151" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.sei.EndpointMethodHandlerImpl" file="EndpointMethodHandlerImpl.java" line="265" method="invoke" /> <ns2:frame class="com.sun.xml.ws.server.sei.SEIInvokerTube" file="SEIInvokerTube.java" line="100" method="processRequest" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="604" method="__doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="563" method="__doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="548" method="doRun" /> <ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="445" method="runSync" /> <ns2:frame class="com.sun.xml.ws.server.WSEndpointImpl\$2" file="WSEndpointImpl.java" line="373" method="process" /> <ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter\$HttpToolkit" file="HttpAdapter.java" line="524" method="handle" /> <ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter" file="HttpAdapter.java" line="255" method="handle" /> <ns2:frame class="com.sun.xml.ws.transport.http.servlet.ServletAdapter" file="ServletAdapter.java" line="140" method="handle" /> <ns2:frame class="weblogic.wsee.jaxws.WLSServletAdapter" file="WLSServletAdapter.java" line="208" method="handle" /> <ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter\$AuthorizedInvoke" file="HttpServletAdapter.java" line="310" method="run" /> <ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter" file="HttpServletAdapter.java" line="223" method="post" /> <ns2:frame class="weblogic.wsee.jaxws.JAXWServlet" file="JAXWServlet.java" line="124" method="doPost" /> <ns2:frame class="javax.servlet.http.HttpServlet" file="HttpServlet.java" line="727" method="service" /> </ns2:stackTrace> </ns2:exception> </detail> </S:Fault> </S:Body> </S:Envelope></pre>	

PASO 2

Creamos un nuevo proyecto llamado PublicacionSubscripcion. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

En la carpeta Recursos añadimos el wsdlPocho.



Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso* http://localhost:8080/CanalPuntoConfiablePaso2/Recursos/WSDLPochoPort?WSDL

Nombre del Recurso*

Tipo de Recurso* WSDL ▾

Siguiente >> Cancelar

“Siguiente”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 2 de 2

<input checked="" type="checkbox"/>	Nombre de Archivo ▾	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	http://localhost:8080/CanalPuntoConfiablePaso2/Recursos/WSDLPochoPort?WSDL	WSDLPochoPort	CanalPuntoConfiablePaso2/Recursos	WSDL
<input checked="" type="checkbox"/>	http://localhost:8080/CanalPuntoConfiablePaso2/Recursos/XMLSchema_1159110265	XMLSchema_1159110265	CanalPuntoConfiablePaso2/Recursos	Esquema XML

Elementos 1 - 2 de 2

<< Anterior Importar Cancelar

“Importar”.

Creamos en la carpeta *Business Services* dos servicios de negocio a partir del wsdl que hemos creado (BusinessService1 y BusinessService2). De esta manera tenemos dos instancias del mismo servicio.

Referencias	Seleccionar Tipo de Recurso	ón
	Servicio	pción -
	Servicio de Proxy	
Referencias de otro	Servicio de Negocio	Descripción
	División-Unión	
Carpetas	Interfaz	
	WSDL	
Introducir Nom	Esquema XML	
	Política de Servicios Web	
	Enlace de JCA	Agregar Carpeta
	Transformación	
	XQuery	
Nombre	XSLT	
	Archivo MFL	
	Seguridad	
	Cuenta de Servicio	No hay ninguna carpeta que mostrar.
	Proveedor de Claves de Servicio	
	Utilidad	
	JAR	
Suprimir	Destino de Alerta	
	Documento XML	
Recursos	Bloque	
	Recursos de URL	
	Recursos Comprimidos	
Crear Recurso:	Seleccionar Tipo de Recurso	

Ponemos un nombre y seleccionamos el wsdl que acabamos de crear en la carpeta recursos. “Siguiente”.

Crear un Servicio de Negocio (CanalPuntoAPuntoConfiablePaso2/Business Services/)

Configuración General

Nombre del Servicio* BusinessService1

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

CanalPuntoAPuntoConfiablePaso2/Recursos/WSDLPochoPort

Examinar...

WSDLPochoPort (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

Examinar...

☐ Servicio de Proxy

Examinar...

Siguiente >> Último >> Cancelar

Seleccionamos el protocolo http (está por defecto) y añadimos una URI de punto final. Dejamos el resto de campos como están y pulsamos “último”.

Crear un Servicio de Negocio (CanalPuntoAPuntoConfiablePaso2/Business Services/BusinessService11)

Configuración de Transporte

Protocolo* http

Algoritmo de Equilibrio de Carga

asignación en rueda (round-robin)

URI de Punto Final*

Formato: http://host:port/someService

http://localhost:7001/CanalPuntoAPuntoConfiablePaso2/Business_Services/8

Agregar

URI EXISTENTES

http://**...*/Application4-WSDLUnidireccionalPocho-context-root/WSDLPochoPort

OPCIONES

Recuento de Reintentos

0

Intervalo de Iteración de Reintentos

30

Reintentar tras Errores de Aplicación

☒ Sí ☐ No

<< Anterior Siguiente >> Último >> Cancelar

A continuación creamos el PSCola1.

Le damos un nombre: PSCola1 y lo creamos a partir de cualquier servicio SOAP. “Siguiendo”.

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola que va a utilizar. Podemos cambiar el nombre de la cola. Las colas se crean de manera automática una vez que ejecutemos el Proxy.

“Siguiendo”.

Seleccionamos la siguiente configuración de transporte:

Editar un Servicio de Proxy (PublicacionSubscripcion/Proxy Services/PSCola1)	
Configuración de Transporte de JMS	
Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema
Se Necesita Respuesta	<input type="checkbox"/>
Política de Distribución	default
Codificación de Peticiones	UTF-8
Cuenta de Servicio de JMS	<input type="text"/> <input type="button" value="Examinar..."/>
Configuración Avanzada	
<input type="button" value=" << Anterior"/> <input type="button" value=" Siguiente >>"/> <input type="button" value=" Último >>"/> <input type="button" value=" Cancelar"/>	

En este ejemplo al ser un servicio de entrada no se necesita respuesta.

Desplegamos Configuración avanzada.

Introducimos el número de reintentos en caso de fallo, el tiempo entre cada uno de esos reintentos (en milisegundos), el destino de error (en este caso la cola ColaError) y en caso de que después de todos los reintentos siga fallando se redirija a esta cola.

Configuración Avanzada	
Usar SSL	<input type="checkbox"/>
Selector de Mensajes	<input type="text"/>
Suscripción Duradera	<input type="checkbox"/>
Recuento de Reintentos	2
Intervalo de Reintentos	5000
Destino de Error	ColaErrorSubscriber1
Política de Caducidad	Redireccionar
Se necesita XA	<input type="checkbox"/>
<input type="button" value=" << Anterior"/> <input type="button" value=" Siguiente >>"/> <input type="button" value=" Último >>"/> <input type="button" value=" Cancelar"/>	

“Siguiente”.

“Siguiente”. Seleccionamos misma transacción para respuesta. Si algo ocurre se produce rollback en la transacción.

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)	
Manejo de Mensajes	
Transacción Necesaria	<input type="checkbox"/> Activado
Misma Transacción para Respuesta	<input checked="" type="checkbox"/> Activado
Flujo de Contenido	<input type="checkbox"/> Activado Tipo de Buffer <input type="radio"/> Buffer de Memoria <input type="radio"/> Buffer de Disco Compresión <input type="checkbox"/> Activado
<input type="button" value=" << Anterior"/> <input type="button" value=" Siguiente >>"/> <input type="button" value=" Último >>"/> <input type="button" value=" Cancelar"/>	


“Último” y “Guardar”.

A continuación creamos PSSubscriptor1. (PSSubscriptor2 es igual pero sin crear el tema).

Le damos un nombre (PSSubscriptor1) y lo creamos a partir de cualquier servicio SOAP:

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola/tema que va a utilizar. Podemos cambiar el nombre de la cola. Las colas se crean de manera automática una vez que ejecutemos el Proxy. Todos los proxys subscriptores deben tener la misma URI para que una vez que se publique el mensaje en dicho tema, todos tengan acceso a él.

En este caso no va a ser una cola, sino un tema. Sólo en este subscriptor creamos una cola de errores, ya que es el encargado de crear el tema y la cola de errores. El resto de subscriptores sólo harán referencia al tema.

 Editar un Servicio de Proxy (PublicacionSubscripcion/Proxy Services/PSSubscriptor1)

Configuración de Transporte de JMS


Tipo de Destino	<input type="radio"/> Cola <input checked="" type="radio"/> Tema
Se Necesita Respuesta	<input type="checkbox"/>
Política de Distribución	default ▼
Codificación de Peticiones	UTF-8
Cuenta de Servicio de JMS	<input type="text"/> <input type="button" value="Examinar..."/>

Configuración Avanzada

Usar SSL	<input type="checkbox"/>
Selector de Mensajes	<input type="text"/>
Suscripción Duradera	<input checked="" type="checkbox"/>
Recuento de Reintentos	2
Intervalo de Reintentos	5000
Destino de Error	ColaErroPublicador
Política de Caducidad	Redireccionar ▼
Se necesita XA	<input type="checkbox"/>

“Siguiente”.

Misma transacción para respuesta. “Siguiente”.







 Editar un Servicio de Proxy (PublicacionSubscripcion/Proxy Services/PSSubscriptor1)

Manejo de Mensajes

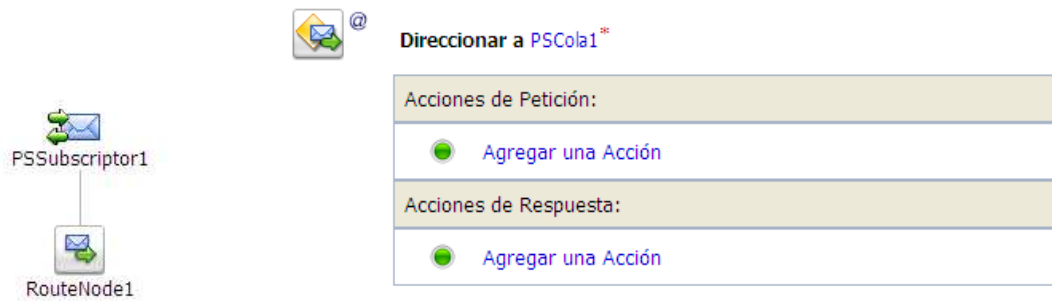
Transacción Necesaria	<input type="checkbox"/> Activado
Misma Transacción para Respuesta	<input checked="" type="checkbox"/> Activado
Flujo de Contenido	<input type="checkbox"/> Activado Tipo de Buffer <input type="radio"/> Buffer de Memoria <input type="radio"/> Buffer de Disco Compresión <input type="checkbox"/> Activado

“Guardar”.

Entramos al flujo del mensaje de PSSubscriptor1:

<input type="checkbox"/>  PSSubscriptor1	Servicio de Proxy  	aJe   
---	---	---

Añadimos una nueva acción “Direccionamiento” a PSCola1.




Repetimos los mismo pasos para PSCola2.

Entramos en el flujo de mensajes del PSCola1 y PSCola2 que acabamos de crear. “Editar direccionamiento” y añadimos “Opciones de direccionamiento”. Seleccionamos Calidad del Servicio “Una Sola Vez”, para evitar que se dupliquen los mensajes.

The configuration window titled **Direccionar a BusinessService1** is shown, calling **IntroducirNombre**. It includes a checkbox ☐ **Utilice la operación entrante para la saliente**. Below, the **Acciones de Petición:** section contains a sub-section **Opciones de Direccionamiento** for replacing default configurations. It lists several options with checkboxes: **URI:** ☐ **<Expresión>**, **Calidad del Servicio:** ☒ **Una Sola Vez**, **Modo:** ☐ **Petición**, **Intervalo de Reintentos:** **(segundo)**, **Recuento de Reintentos:** , and **Prioridad:** ☐ **<Expresión>**. The **Acciones de Respuesta:** section at the bottom has a button **Agregar una Acción**.

A continuación creamos el Proxy publicador al que se accede por http.

Le damos un nombre (PSPublicador) y lo creamos a partir del wsdlPocho que hemos creado. “Siguiendo”.

 **Crear un Servicio de Proxy (PublicacionSubscripcion/Proxy Services/)**

Configuración General

Nombre del Servicio* PSPublicador

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL Subscription/Recursos/WSDLPochoPort WSDLPochoPort (puerto) Examinar...

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML


Crear a partir de Servicio Existente

☐ Servicio de Negocio Examinar...

☐ Servicio de Proxy Examinar...

Siguiente >> Último >> Cancelar

Protocolo de transporte HTTP. Pulsamos “siguiente” hasta el final sin hacer ningún cambio más.

 **Crear un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS11)**

Configuración de Transporte

Protocolo* http

URI de Punto Final* Formato: /someName
/CanalPuntoAPuntoConfiablePaso2/Proxy_Services/PS1











Obtener Todas las Cabeceras

☒ Sí

☐ No

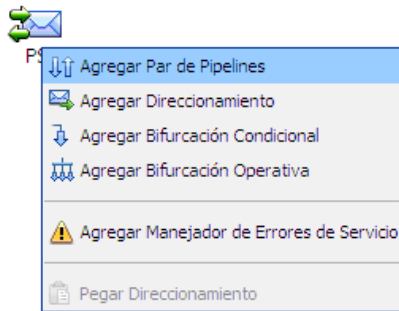
<< Anterior Siguiente >> Último >> Cancelar

Por último vamos al flujo de mensajes del PSPublicador.

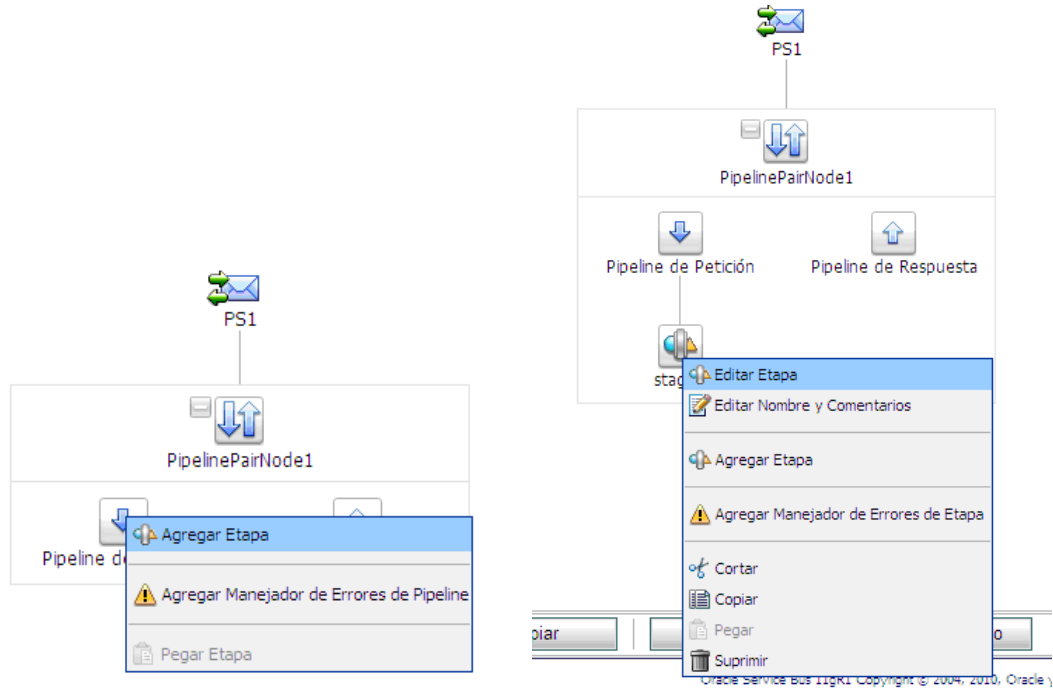
Recursos			
▶ Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 5 de 5			
<input type="checkbox"/> Nombre	Tipo de Recurso	Acciones	Opciones
<input type="checkbox"/> PSCola1	Servicio de Proxy		a e 
<input type="checkbox"/> PSCola2	Servicio de Proxy		a e 
<input type="checkbox"/> PSPublicador	Servicio de Proxy		a e 
<input type="checkbox"/> PSSubscriptor1	Servicio de Proxy		a e 
<input type="checkbox"/> PSSubscriptor2	Servicio de Proxy		a e 
Elementos 1 - 5 de 5			
Suprimir			

Añadimos un pipeline. Como queremos mandar el mensaje pero no queremos esperar la respuesta sino obtener una respuesta inmediata debemos utilizar “Publish” en vez de “Route to”. Por ello:

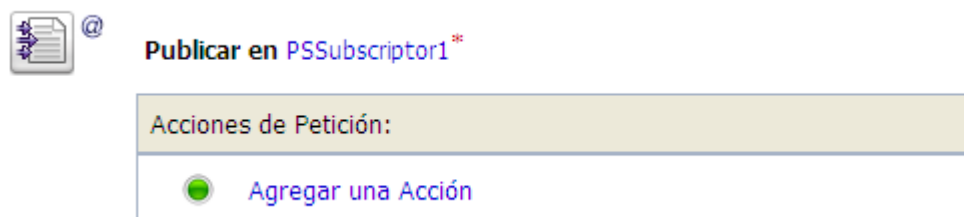
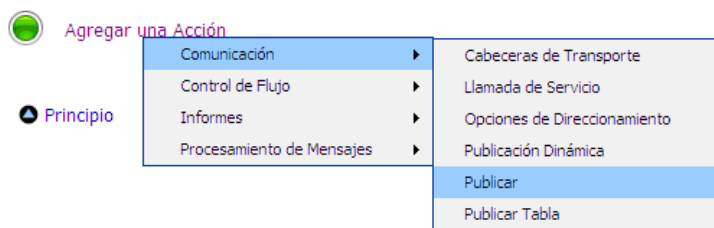
9. Añadimos un par de pipelines



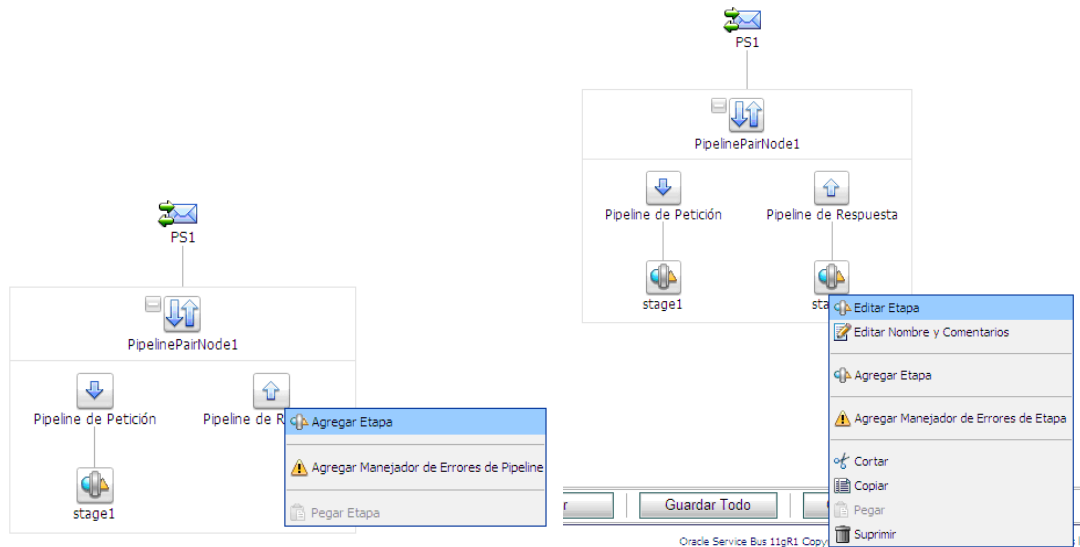
10. En el pipeline de petición añadimos una nueva etapa.



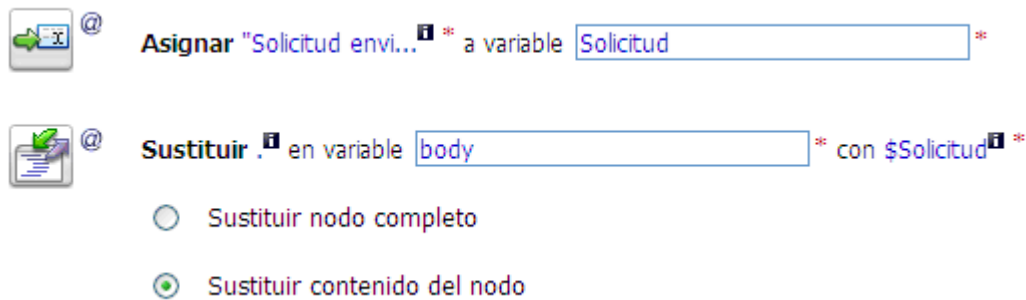
11. Añadimos una nueva acción "Publicar". Seleccionamos al servicio al que queremos direccionar, en este caso PSSubscriptor1.



12. En el pipeline de respuesta añadimos una nueva etapa.



13. Añadimos una acción “Asignar” para crear la respuesta del Proxy.



Prueba en la consola del OSB

Ejecutamos varias veces el PSPublicador.

Recursos				
Crear Recurso: Seleccionar Tipo de Recurso				
Elementos 1 - 5 de 5				
Nombre	Tipo de Recurso	Acciones	Opciones	
PSCola1	Servicio de Proxy		a e	
PSCola2	Servicio de Proxy		a e	
PSPublicador	Servicio de Proxy		a e	
PSSubscriptor1	Servicio de Proxy		a e	
PSSubscriptor2	Servicio de Proxy		a e	
Elementos 1 - 5 de 5				
Suprimir				

Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación.

Prueba de Servicios de Proxy - PSPublicador Ayuda

Atrás Cerrar

Documento de Petición

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <wsd:IntroducirNombre xmlns:wsd="http://wsdlunidireccionalpocho/">
      <!--Optional-->
      <arg0>string</arg0>
    </wsd:IntroducirNombre>
  </soapenv:Body>
</soapenv:Envelope>
```

Documento de Respuesta

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>Solicitud enviada</soapenv:Body>
</soapenv:Envelope>
```

Metadatos de Respuesta

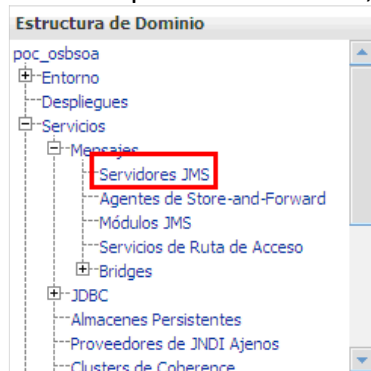
```
<con:metadata xmlns:con="http://www.bea.com/wli/sb/test/config">
  <tran:headers xsi:type="http:HttpResponseHeaders" xmlns:http="http://www.bea.com/wli/sb/transports/http"
    xmlns:tran="http://www.bea.com/wli/sb/transports" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <http:Content-Type>text/xml; charset=utf-8</http:Content-Type>
  </tran:headers>
  <tran:response-code xmlns:tran="http://www.bea.com/wli/sb/transports">0</tran:response-code>
  <tran:encoding xmlns:tran="http://www.bea.com/wli/sb/transports">utf-8</tran:encoding>
</con:metadata>
```

Rastreo de Llamada

(recibiendo petición)

PipelinePairNode1

Para comprobar si funciona, vamos al weblogic. Servicios / Mensajes / Servidores JMS



Elegimos el servidor wlsbJMSServer.

Resumen de Servidores JMS

Los servidores JMS actúan como contenedores de gestión para las colas y los temas en los módulos JMS que están dirigidos a ellos.

En esta página se resumen los servidores JMS que se han creado en el dominio de WebLogic Server actual.

[Personalizar esta Tabla](#)

Servidores JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Almacén Persistente	Destino	Servidor Actual	Estado
<input type="checkbox"/>	BPMJMServer	BPMJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	JMSserver	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	SOAJMServer	SOAJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	UMSJMServer	UMSJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	wlsbJMServer	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	WseeJmsServer	WseeFileStore	osb_server1	osb_server1	✓ OK

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

Pestaña Supervisión / Destinos Activos.

En este ejemplo, se ha ejecutado el PSPublicador 6 veces, de las cuales 4 han tenido éxito y 2 han fallado.

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

Mostrando 1 a 10 de 13 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!ColaErrorPublicador	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ColaErrorSubscriber1	2	0	2	2	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!ColaErrorSubscriber2	2	0	2	2	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PeticiónPubSub-811781088	0	0	1	6	0	Topic	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PSCola1Petición-1244462166	0	0	2	6	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PSCola2Petición1077597518	0	0	2	6	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

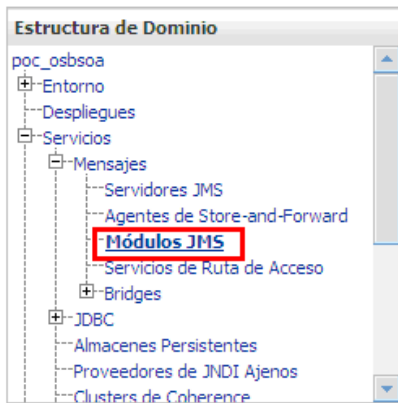
Producción Consumo Inserción

Mostrando 1 a 10 de 13 Anterior | Siguiente

PeticiónPubSub→6 mensajes de entrada.

Estos 6 mensajes de entrada se duplican en PSCola1Petición y PSCola2Petición.

Si entramos en Servicios / Módulos JMS, podemos ver los mensajes que se han guardado en las colas de errores.



Módulo jmsResources.

Módulos JMS

Los recursos del sistema de JMS se configuran y almacenan como módulos parecidos a los módulos de J2EE estándar. Dichos recursos incluyen colas, temas, fábricas de conexiones, plantillas, claves de destino, cuotas, colas distribuidas, temas distribuidos, servidores ajenos y parámetros de Store-and-Forward (SAF) de JMS. Puede configurar y gestionar de forma administrativa módulos de sistema de JMS como recursos del sistema globales.

En esta página se resumen los módulos del sistema de JMS creados para este dominio.

[Personalizar esta Tabla](#)

Módulos JMS

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/> Nombre	Tipo
<input type="checkbox"/> BPMJMSModule	Sistema
<input type="checkbox"/> jmsResources	Sistema
<input type="checkbox"/> SOAJMSModule	Sistema
<input type="checkbox"/> SystemModule	Sistema
<input type="checkbox"/> UMSJMSSystemResource	Sistema
<input type="checkbox"/> WseeJmsModule	Sistema

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

Nombre de la cola: ColaErrorSubscriber1 y ColaErrorSubscriber2.

Resumen de Recursos

Nuevo Suprimir

Mostrando 1 a 10 de 16 Anterior | Siguiente

<input type="checkbox"/> Nombre	Tipo	Nombre de JNDI	Despliegue Secundario	Destinos
<input type="checkbox"/> ColaErrorPublicador	Cola	ColaErrorPublicador	ColaErrorPublicador	wlsbJMSServer
<input type="checkbox"/> ColaErrorSubscriber1	Cola	ColaErrorSubscriber1	ColaErrorSubscriber1	wlsbJMSServer
<input type="checkbox"/> ColaErrorSubscriber2	Cola	ColaErrorSubscriber2	ColaErrorSubscriber2	wlsbJMSServer
<input type="checkbox"/> PeticionPublSub-811781088	Tema	PeticionPublSub	PeticionPublSub-811781088	wlsbJMSServer
<input type="checkbox"/> PSCola1Petición-1244462166	Cola	PSCola1Petición	PSCola1Petición-1244462166	wlsbJMSServer
<input type="checkbox"/> PSCola2Petición1077597518	Cola	PSCola2Petición	PSCola2Petición1077597518	wlsbJMSServer
<input type="checkbox"/> TemporaryTmplt	Plantilla	N/A	N/A	N/A
<input type="checkbox"/> weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Fábrica de Conexiones	weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/> wli.reporting.jmsprovider.ConnectionFactory	Fábrica de Conexiones	wli.reporting.jmsprovider.ConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/> wli.reporting.jmsprovider.queue	Cola	wli.reporting.jmsprovider.queue	wlsbJMSServer	wlsbJMSServer

Nuevo Suprimir

Mostrando 1 a 10 de 16 Anterior | Siguiente

Pestaña Supervisión.

Seleccionamos la cola y pulsamos “Mostrar Mensajes”.

Valores para ColaErrorSubscriber1

Configuración **Supervisión** Control Seguridad Despliegue Secundario Notas

Un destino de JMS identifica una cola (punto a punto) o un tema (publicación/suscripción) dirigidos a un servidor JMS.
En esta página se resumen los destinos de JMS activos que se han creado para este módulo JMS.

[Personalizar esta Tabla](#)

Destinos (Filtrado: Existen Más Columnas)

[Mostrar Mensajes](#) Mostrando 1 a 1 de 1 Anterior | Siguiente

<input checked="" type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Mensajes Totales	Consumidores Actuales	Recuento Alto de Consumidores	Consumidores Totales	Recuento Alto de Mensajes
<input checked="" type="checkbox"/>	jmsResources! ColaErrorSubscriber1	2	0	2	0	0	0	2

[Mostrar Mensajes](#) Mostrando 1 a 1 de 1 Anterior | Siguiente

En este caso se han producido dos mensajes erróneos, en el segundo 00 y 02. Se puede ver el contenido de dichos mensajes.

Resumen de Mensajes de JMS

En esta página se resumen los mensajes disponibles para una cola autónoma, una cola distribuida o un suscriptor duradero de temas. Utilice esta página para ver detalles de mensajes, crear nuevos mensajes, suprimir mensajes seleccionados, mover mensajes a otro destino, exportar contenidos de mensajes en formato XML a otro archivo, importar contenido de mensajes con formato XML de otro archivo o vaciar todos los mensajes de un destino.
Haga clic en un mensaje para ver su contenido.

Selector de Mensajes: [Aplicar](#)

[Personalizar esta Tabla](#)

Mensajes de JMS (Filtrado: Existen Más Columnas)

[Nuevo](#) [Suprimir](#) [Mover](#) [Importar](#) [Exportar](#) Mostrando 1 a 2 de 2 Anterior | Siguiente

<input type="checkbox"/>	Identificador	Identificador de Correlación	Registro de Hora	Cadena de Estado	Modo de Entrega de JMS	Tamaño del Mensaje
<input type="checkbox"/>	ID: <77555.1293183000610.0>		Fri Dec 24 10:30:00 CET 2010	visible	Persistente	381
<input type="checkbox"/>	ID: <77555.1293183002628.0>		Fri Dec 24 10:30:02 CET 2010	visible	Persistente	381

[Nuevo](#) [Suprimir](#) [Mover](#) [Importar](#) [Exportar](#) Mostrando 1 a 2 de 2 Anterior | Siguiente

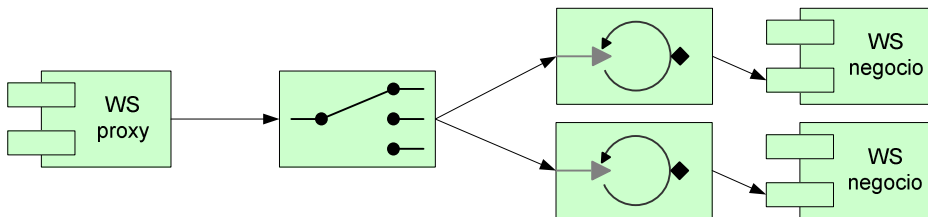
Caso de Uso 4: Encaminamiento basado en el contenido

Paso 1

Desplegaremos dos servicios SOAP de negocio de un solo sentido en el bus. Pueden ser dos instancias del mismo servicio.

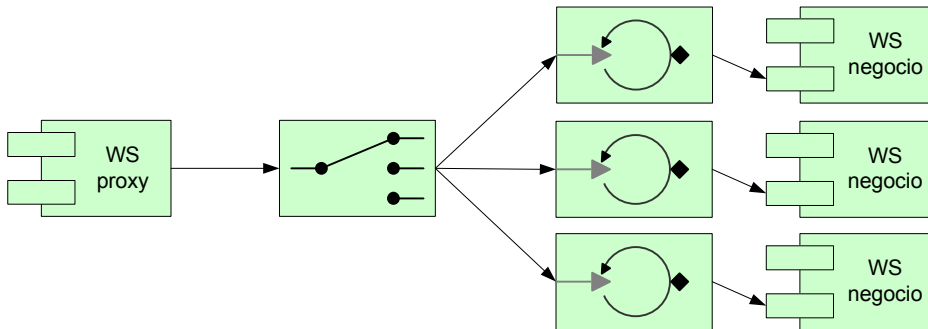
Una aplicación cliente llamará a un solo servicio proxy, con mensajes de diferente contenido.

Entre el proxy y los servicios de negocio se implementará un encaminador que, dependiendo del contenido de los mensajes, los redirigirá a un servicio de negocio u otro.



Paso 2

Se añade un tercer servicio de negocio y se configura el enrutador. La aplicación cliente llamará ahora al proxy con mensajes con tres tipos de contenido diferentes.



Recursos utilizados:

- WSDL

Servicio 1: <http://a.b.c.d:8081/WSBasadoContenido-WSServicio1-context-root/Servicio1SOAP?WSDL>

Servicio 2: <http://a.b.c.d:8081/WSBasadoContenido-WSServicio2-context-root/Servicio2SOAP?WSDL>

Servicio 3: <http://a.b.c.d:8081/WSBasadoContenido-WSServicio33-context-root/Servicio3SOAP?WSDL>

Creamos una sesión en OSB y creamos un nuevo proyecto llamado EncaminamientoBasadoEnContenido2. Dentro de dicho proyecto creamos tres carpetas: *Business Services*, *Proxy Services* y *WSDL*.

En la carpeta WSDL añadimos los recursos necesarios. En este caso los tres WSDL referentes a los servicios de negocios. Para ello pinchamos en:



Añadimos los tres WSDL de la siguiente forma:

- Introducimos la URL donde se encuentra dicho WSDL
- Le damos un nombre
- Y seleccionamos el tipo de recurso WSDL

Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

Le damos al botón “importar”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 1 de 1

<input checked="" type="checkbox"/>	Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	http://.../WSBasadoContenido...	Servicio3	EncaminamientoBasadoEnContenidoPaso2	WSDL

Elementos 1 - 1 de 1

El resultado es el siguiente:

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Elementos 1 - 3 de 3

<input type="checkbox"/> Nombre	Tipo de Recurso	Acciones	Opciones
<input type="checkbox"/> Servicio1	WSDL		añe
<input type="checkbox"/> Servicio2	WSDL		añe
<input type="checkbox"/> Servicio3	WSDL		añe

Elementos 1 - 3 de 3

A continuación vamos a crear los respectivos *Business Services*. Para ello vamos a la carpeta *Business Service* y creamos uno nuevo:

Referencias

Referencias de otro

Carpetas

► Introducir Nombre

Nombre

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Seleccionar Tipo de Recurso

Servicio

Servicio de Proxy

Servicio de Negocio

División-Unión

Interfaz

WSDL

Esquema XML

Política de Servicios Web

Enlace de JCA

Transformación

XQuery

XSLT

Archivo MFL

Seguridad

Cuenta de Servicio

Proveedor de Claves de Servicio

Utilidad

JAR

Destino de Alerta

Documento XML

Bloque

Recursos de URL

Recursos Comprimidos

Les damos los nombres Servicio1_BS, Servicio2_BS y Servicio3_BS respectivamente y creamos cada servicio a partir de los WSDL anteriormente añadidos:

Crear un Servicio de Negocio (EncaminamientoBasadoEnContenidoPaso2/Business Services/)

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Una vez que tenemos los WSDL y los *Business Services* creados, procedemos a crear el *Proxy Service* en su respectiva carpeta.



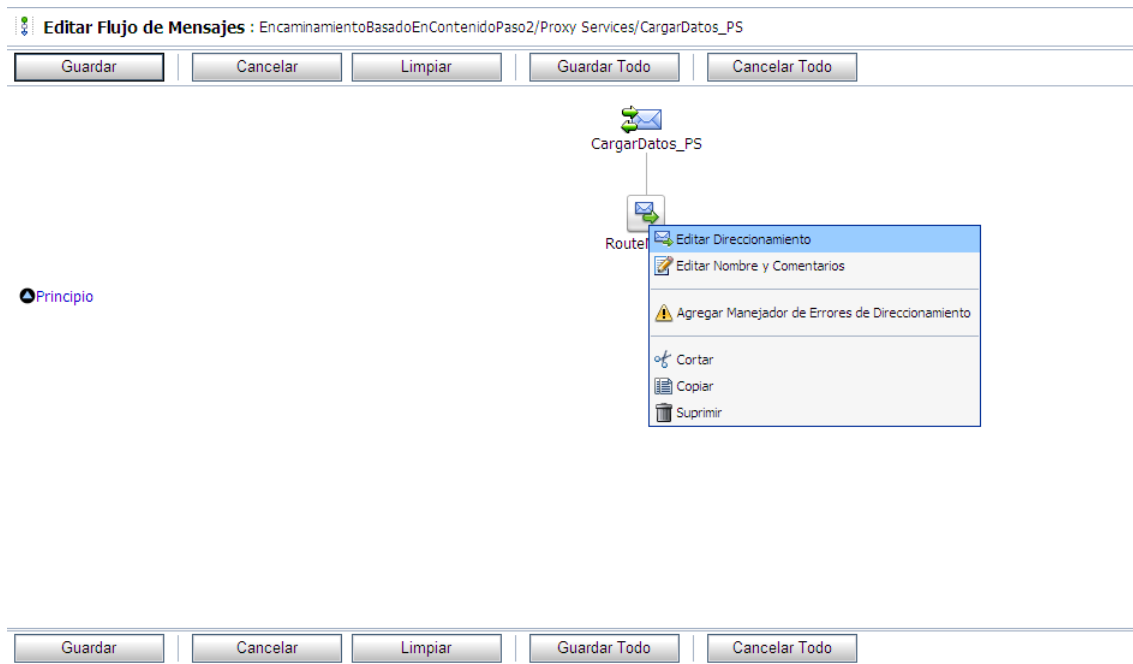
Le damos el nombre CargarDatos_PS y lo creamos a partir del WSDL Servicio1. De esta manera definimos la estructura de datos de entrada y de salida que necesita este proxy.

Crear un Servicio de Proxy (EncaminamientoBasadoEnContenidoPaso2/Proxy Services/)

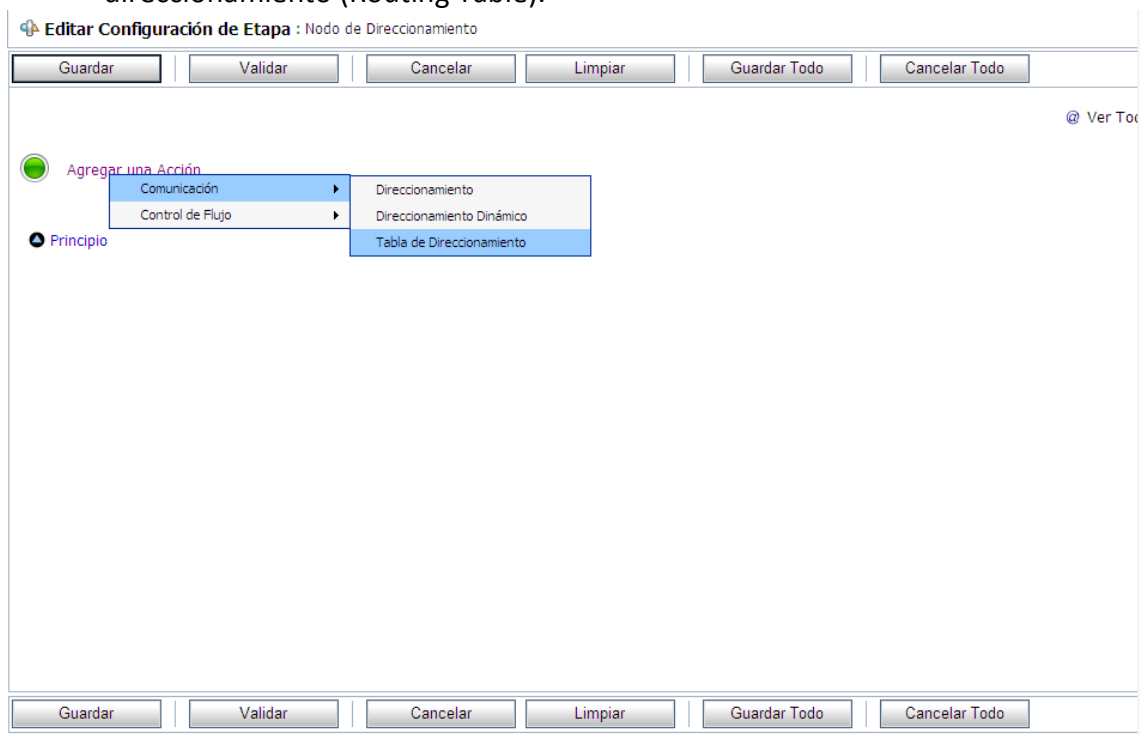
Configuración General	
Nombre del Servicio*	CargarDatos_PS
Descripción	
Tipo de Servicio*	<p>Crear un Servicio Nuevo</p> <p><input checked="" type="radio"/> Servicio Web WSDL <input type="text" value="EncaminamientoBasadoEnContenidoPas"/> <input type="button" value="Examinar..."/></p> <p><input type="radio"/> Servicio de Transporte según Configuración <input type="text" value="Servicio1SOAP"/> (puerto)</p> <p><input type="radio"/> Servicio de Mensajes <input type="text" value="SOAP 1.1"/></p> <p><input type="radio"/> Cualquier Servicio SOAP</p> <p><input type="radio"/> Cualquier Servicio XML</p> <p>Crear a partir de Servicio Existente</p> <p><input type="radio"/> Servicio de Negocio <input type="text"/> <input type="button" value="Examinar..."/></p> <p><input type="radio"/> Servicio de Proxy <input type="text"/> <input type="button" value="Examinar..."/></p>

Una vez creado el proxy, accedemos al flujo de mensajes. Aquí es donde se va a definir el encaminamiento.

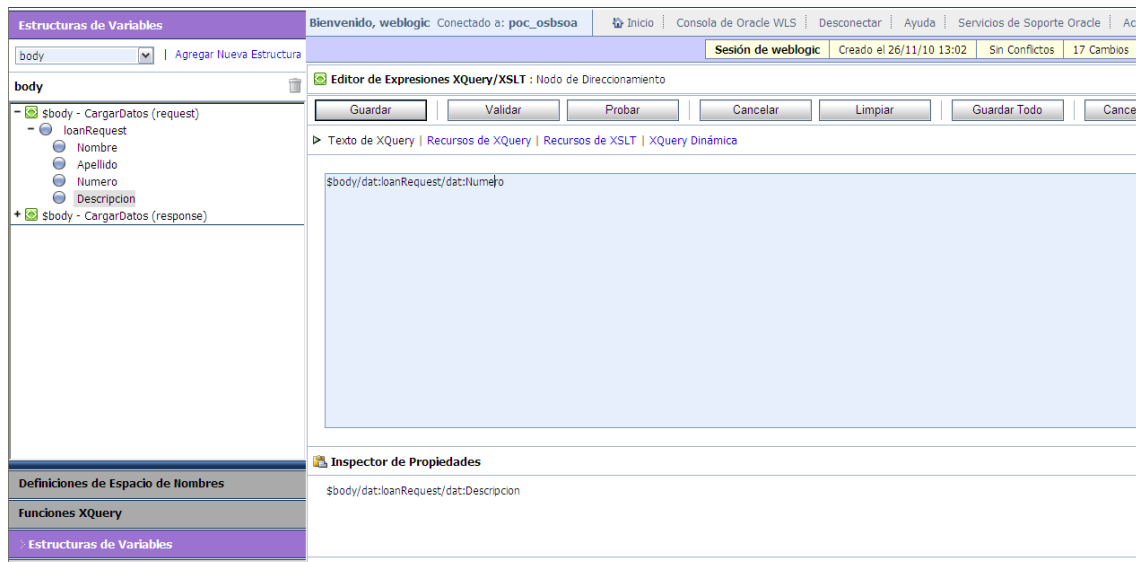
- Pinchamos sobre el proxy y “añadir direccionamiento”.
- Pinchamos sobre el direccionamiento creador y “Editar Direccionamiento”.



- Pinchamos sobre Agregar una acción / Comunicación / Tabla de direccionamiento (Routing Table).

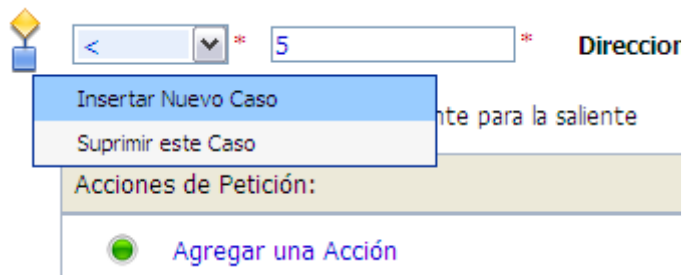


- En el campo expresión : En el panel de la izquierda seleccionamos estructura de variables /body / Numero y arrastramos al panel principal:






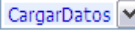
Si el valor de la variable número que acabamos de añadir en la expresión es menor que 5, se llama al Servicio1_BS. Si es igual a 5 se llama al Servicio2_BS y si es mayor que 5 se llama al servicio Servicio3_BS.

- Añadimos tres casos:







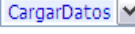
- Hacemos las tres llamadas con las tres condiciones.

 **Tabla de Direccionamiento** (\$body/dat:loanR...¹*)



  * 5 * **Direccionar a Servicio1_BS** llamando a 



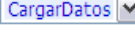
☐ Utilice la operación entrante para la saliente

Acciones de Petición:
 Agregar una Acción
Acciones de Respuesta:
 Agregar una Acción



  * 5 * **Direccionar a Servicio2_BS** llamando a 

☐ Utilice la operación entrante para la saliente

Acciones de Petición:
 Agregar una Acción
Acciones de Respuesta:
 Agregar una Acción

  * 5 * **Direccionar a Servicio3_BS** llamando a 

☐ Utilice la operación entrante para la saliente

Acciones de Petición:
 Agregar una Acción
Acciones de Respuesta:
 Agregar una Acción

Guardamos todo y activamos la sesión.

A continuación hacemos la prueba desde un cliente creado en Java: (ClienteEncaminamientoContenido)

Introducimos los valores. El campo número es el que decide el encaminamiento.

Aplicación Encaminamiento según Contenido

Nombre:

Apellido:

Número: **

Descripción:

**Si el número es <5 llamada al servicio 1, si es =5 llamada al servicio 2 y si es >5 llamada al servicio 3.



Si es menor que 5:

DATOS OBTENIDOS:

Nombre = almudena Apellido = ochoa Número = 3.0 Descripción = Llamada servicio 1

Si es igual que 5:

DATOS OBTENIDOS:

Nombre = almudena Apellido = ochoa Número = 5.0 Descripción = Llamada servicio 2

Si es mayor que 5:

DATOS OBTENIDOS:

Nombre = almudena Apellido = ochoa Número = 7.0 Descripción = Llamada servicio 3

Caso de Uso 5: Separador

Alcance:

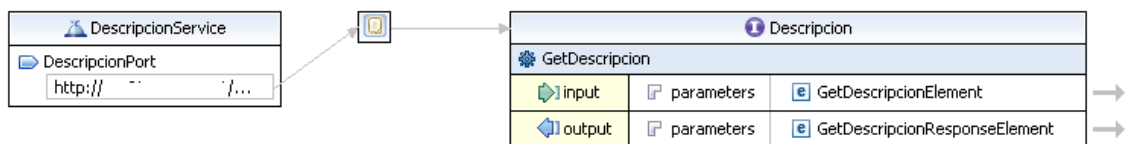
Tenemos un servicio de negocio SOAP de un solo sentido que acepta mensajes de un tipo. La aplicación cliente envía un agregado de mensajes de ese tipo. El proxy acepta el agregado. Desplegamos el servicio en el bus y entre el proxy y el servicio de negocio se implementa un separador que envía al servicio de negocio los mensajes individuales.

Ejemplo desarrollado: introducimos varios ítems. Tratamos cada ítem por separado cogiendo el campo descripción y llamando a un servicio de negocio al cual le pasamos como parámetro de entrada dicho campo descripción.

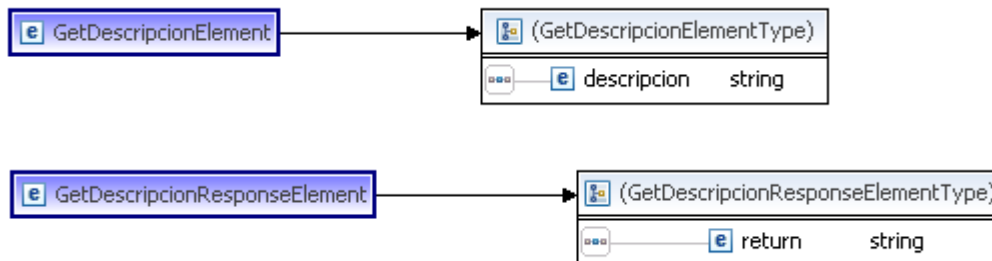
Creamos un nuevo proyecto llamado Separador. Añadimos tres nuevas carpetas: *Business Services*, *Proxy Services* y Recursos.

En la carpeta Recursos añadimos los siguientes wsdl:

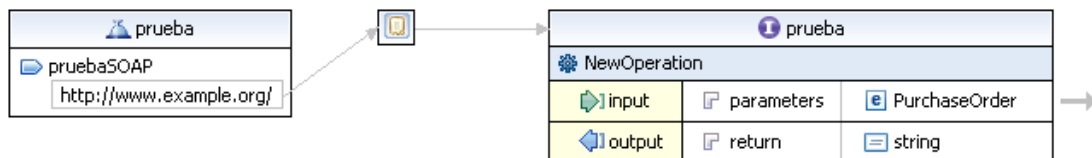
DescripcionPort.wsdl



que sigue el siguiente esquema:

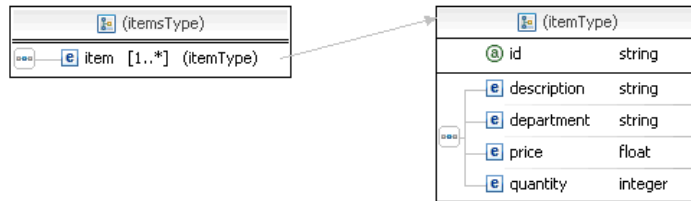


EnviarPetición.wsdl

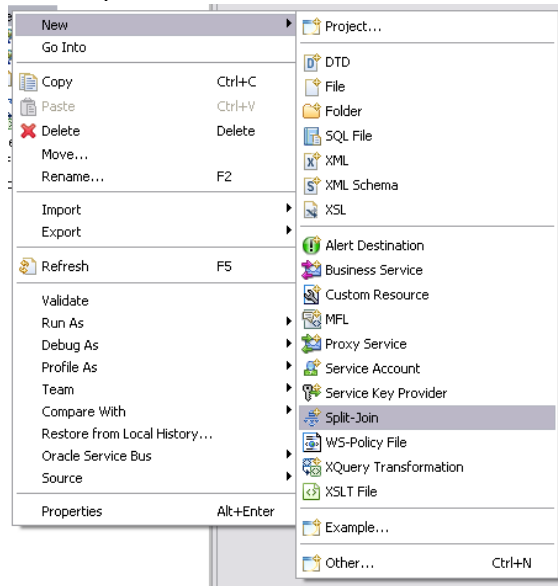


que sigue el siguiente esquema:

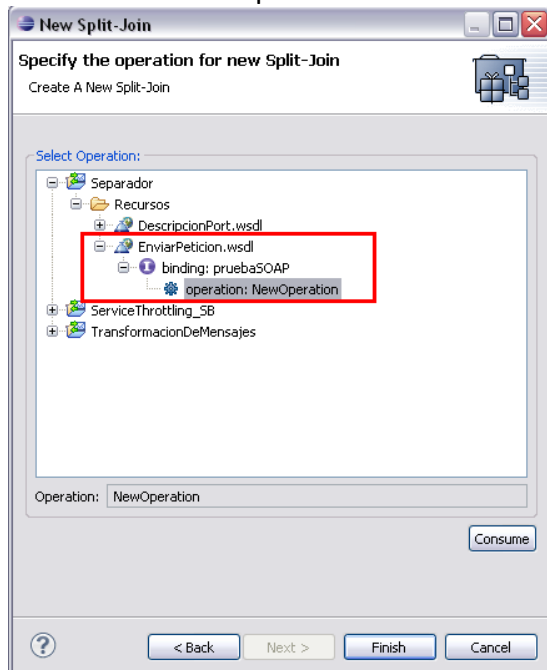




En la carpeta Recursos creamos un nuevo Split-Join y le damos un nombre, en este caso "SplitJoin":

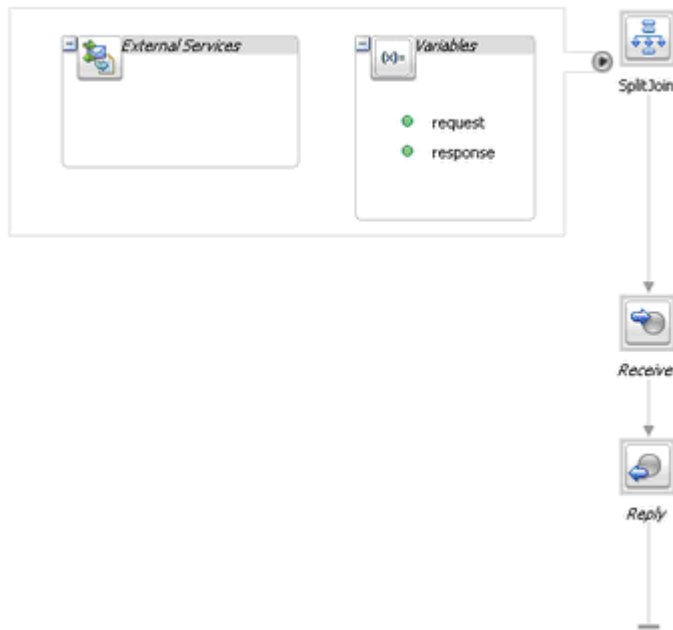


Seleccionamos la operación del WSDL de entrada:

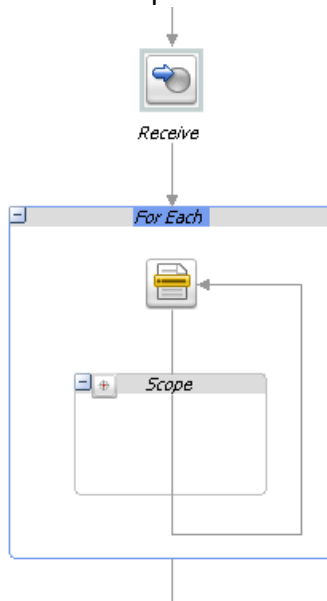


Finish.

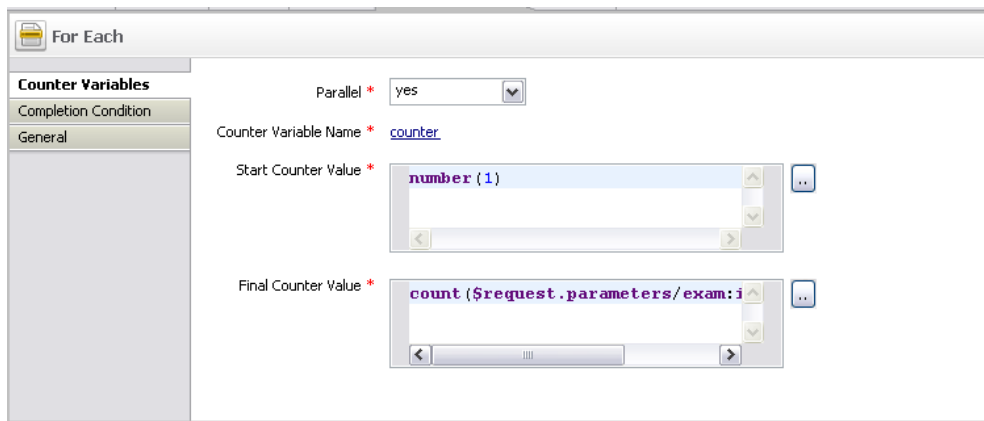
El Split-Join crea automáticamente dos variables: request y response.



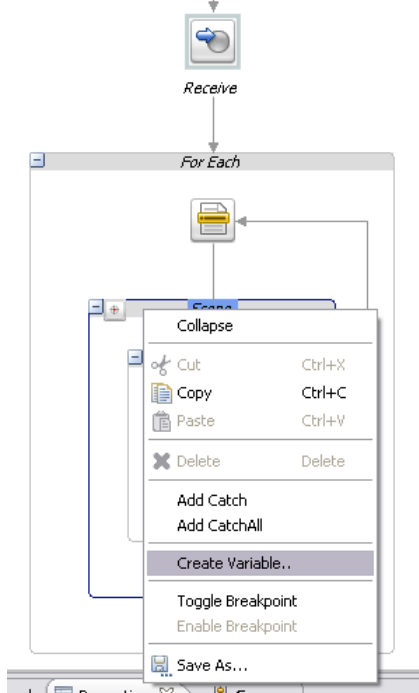
En la variable request se encuentran los datos de entrada. Añadimos una nueva acción “for each” después del “Receive”:



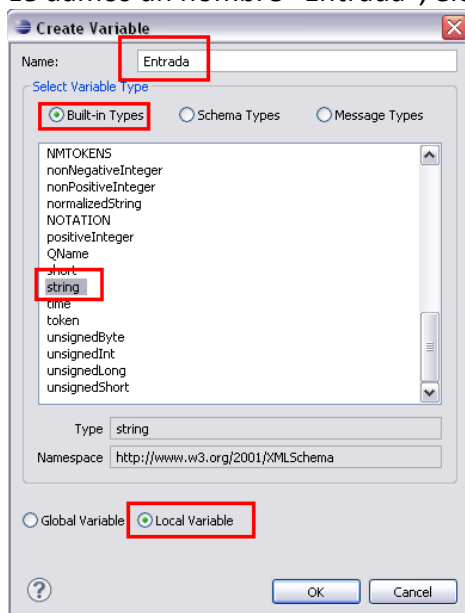
Añadimos propiedades al for-each. En el campo “Counter Variable Name” introducimos un nombre cualquiera para el contador, por ejemplo counter. En el campo “Start Counter Value” indicamos number(1), es decir, desde el primero hasta (“Final Counter Value”) count(\$request.parameters/exam:items/exam:item), es decir, el número total de ítem que existan.



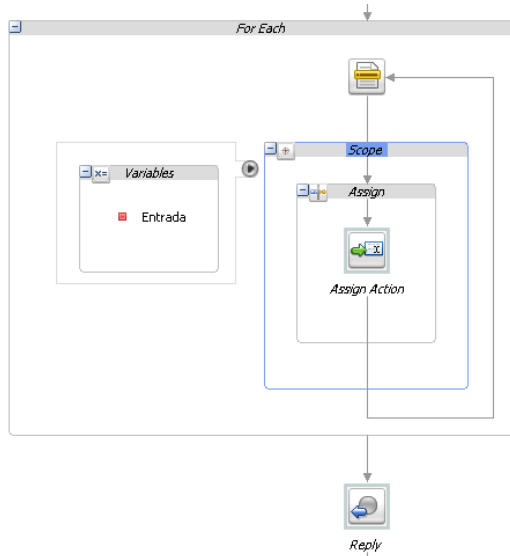
Botón derecho sobre “scope” dentro del for-each y crear variable local:



Le damos un nombre “Entrada”, elegimos el tipo “String” y que sea una variable local.

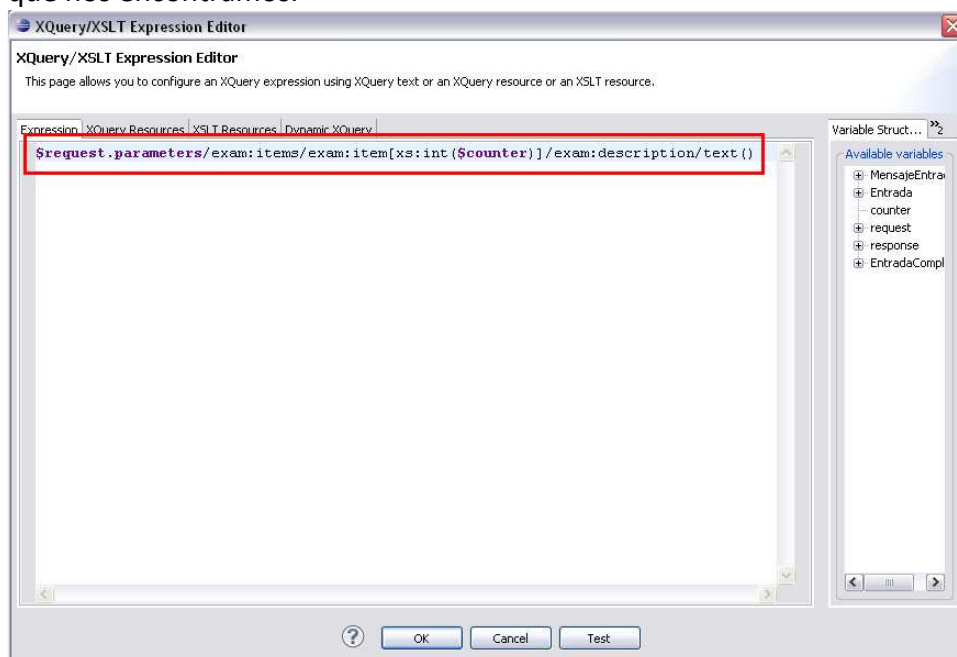


Dentro del “Scope” del for-each añadimos una nueva acción “Assign” la cual nos va a permitir asignar el campo descripción de cada ítem a la variable de tipo String que acabamos de crear.

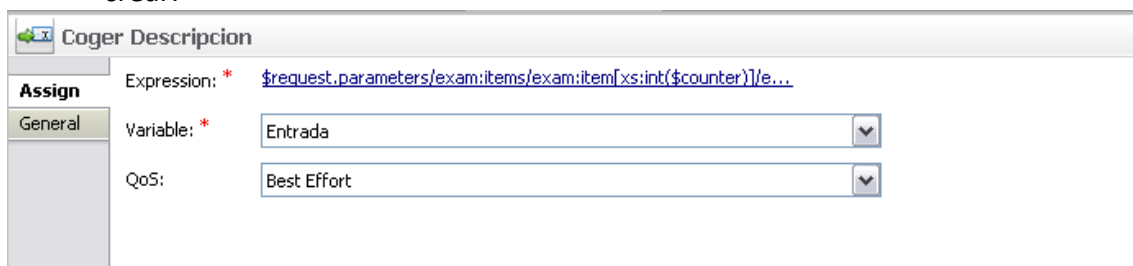


En las propiedades del “Assign” introducimos:

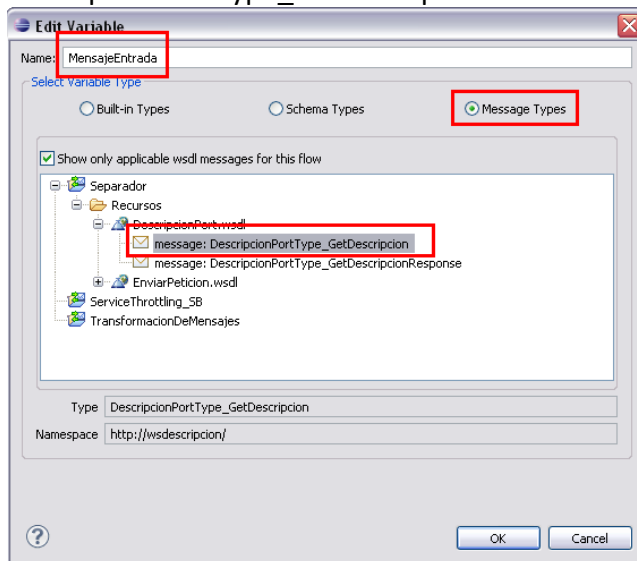
- En el campo “Expression” seleccionamos el campo descripción del ítem en el que nos encontramos.



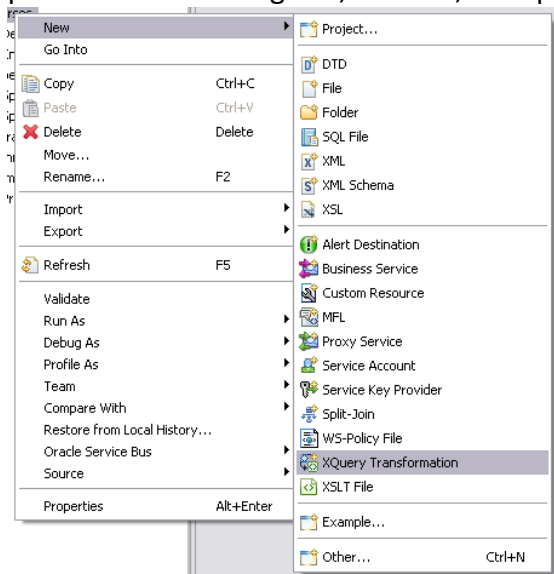
- En el campo “Variable” introducimos la variable local Entrada que acabamos de crear.



Creamos una nueva variable local llamada “MensajeEntrada”. Esta variable va a ser del tipo esperado por el servicio de negocio al que vamos a invocar. En este caso la invocación va a ser al servicio DescriptionPort.wsdl cuyo tipo de datos de entrada es DescriptionPortType_GetDescription.



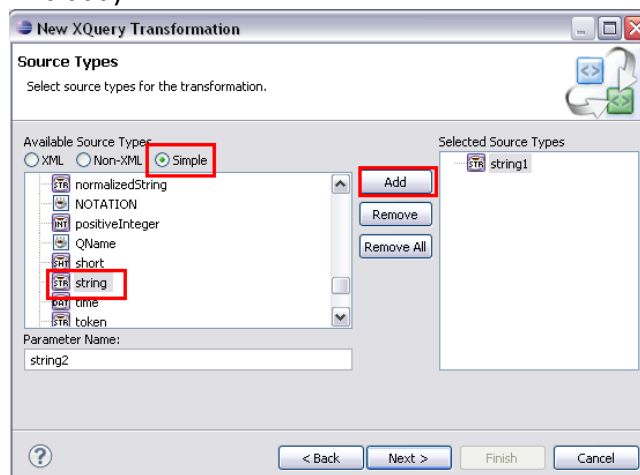
En la carpeta Recursos creamos una nueva transformación xquery que va a ser la encargada de transformar el string de la variable Entrada en el tipo de dato esperado por el servicio de negocio, es decir, del tipo DescriptionPortType_GetDescription:



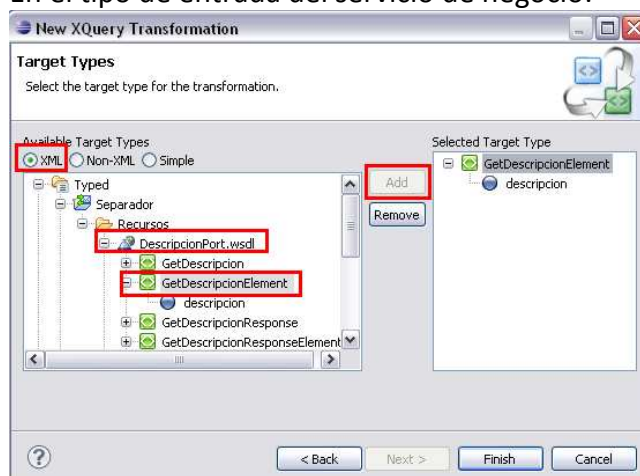
- Añadimos un nombre como por ejemplo “transformar”:



- Queremos transformar un string cualquiera (en nuestro caso la variable Entrada):

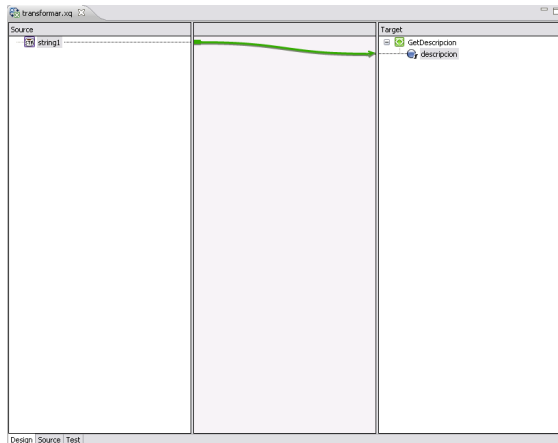


- En el tipo de entrada del servicio de negocio:

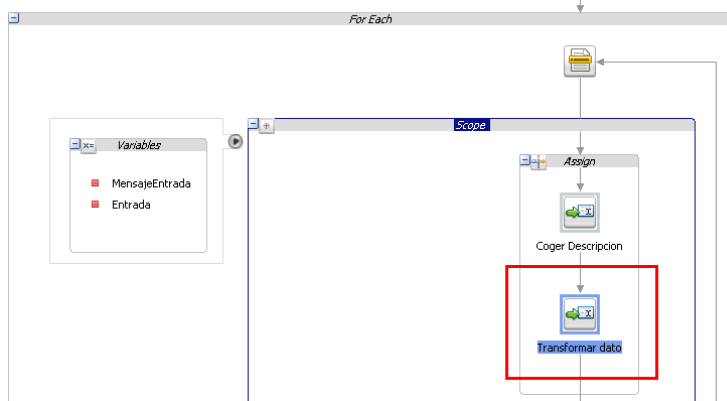


Finish.

- Unimos el string con el campo descripción:

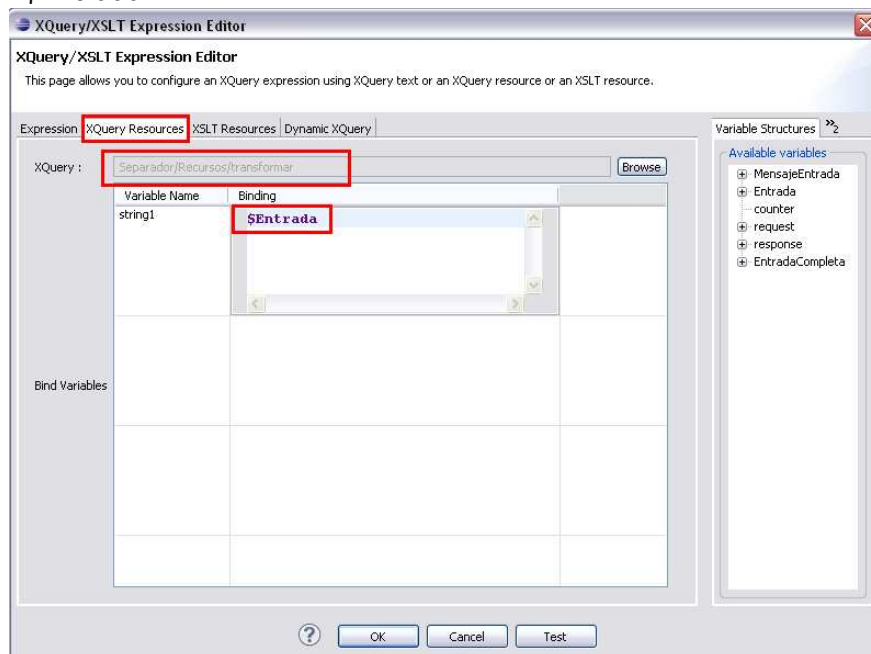


Añadimos un nuevo “Assign” después del que acabamos de configurar. Este Assign va a ser el encargado de realizar la transformación de los datos:

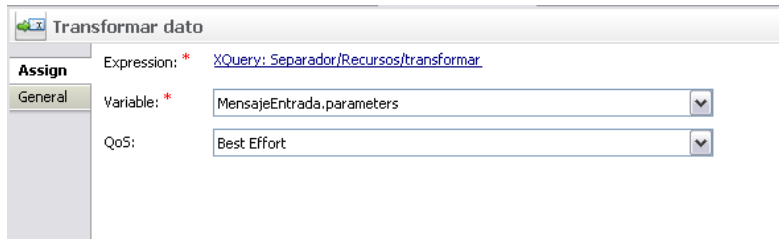


En las propiedades del “Assign” introducimos:

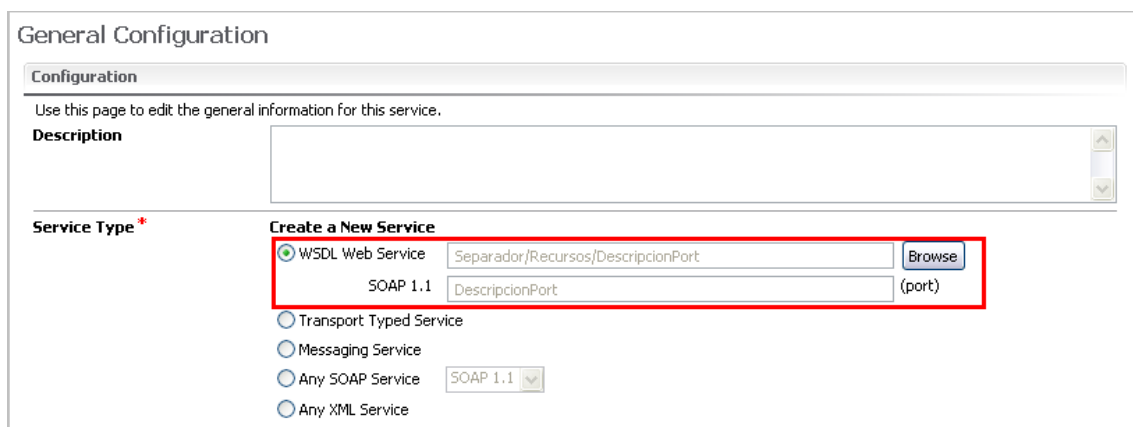
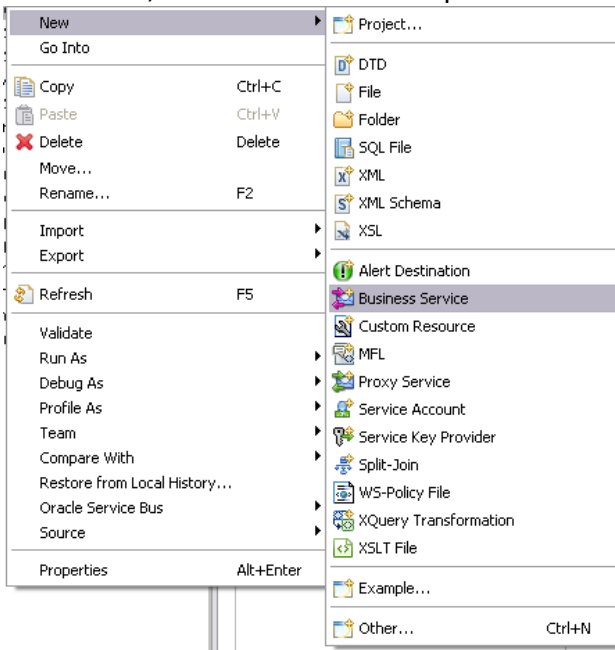
- En el campo “Expression” añadimos la XQuery que hemos creado “transformar.xq” y la variable que queremos transformar, en este caso “\$Entrada”:



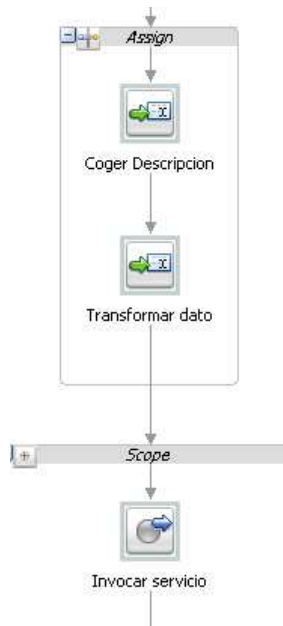
- En el campo “Variable” introducimos la variable local MensajeEntrada que hemos creado.



A continuación creamos en la carpeta *Business Service* el servicio de negocio a partir del *Web Service* que queremos invocar, en este caso *DescripcionPort.wSDL*. Le damos un nombre, en este caso *BSDescripcion*:

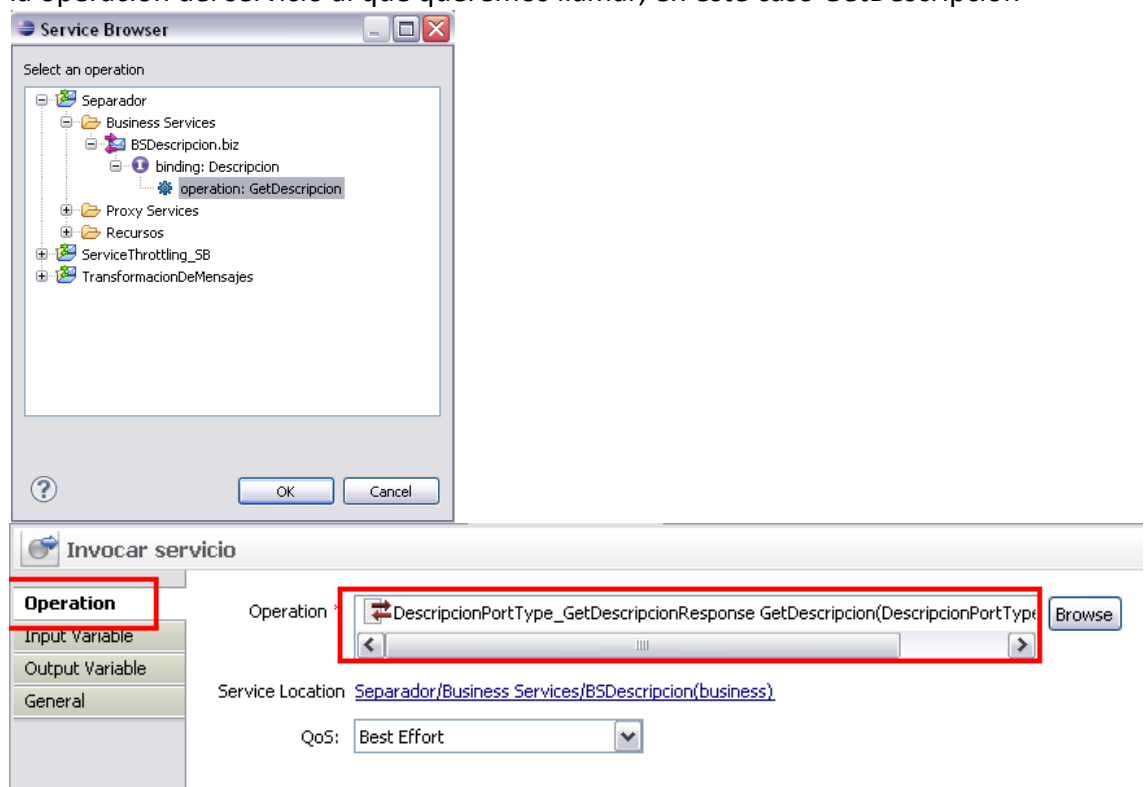


Continuamos con el *SplitJoin*. Añadimos un nuevo “scope” y añadimos la acción “invoke service”:

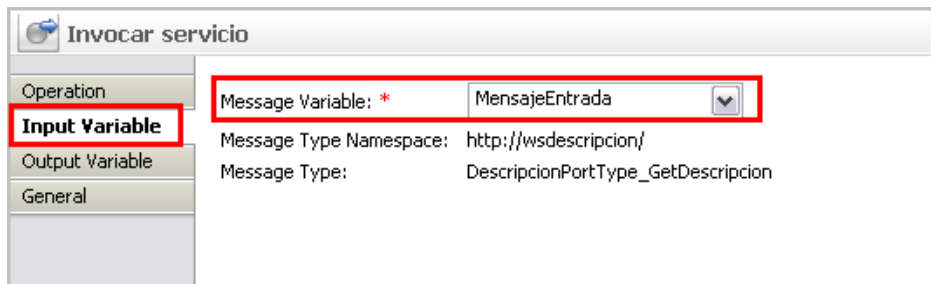


En las propiedades seleccionamos:

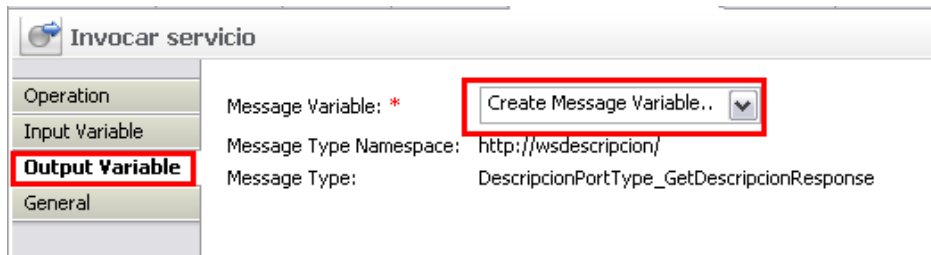
- la operación del servicio al que queremos llamar, en este caso GetDescripcion



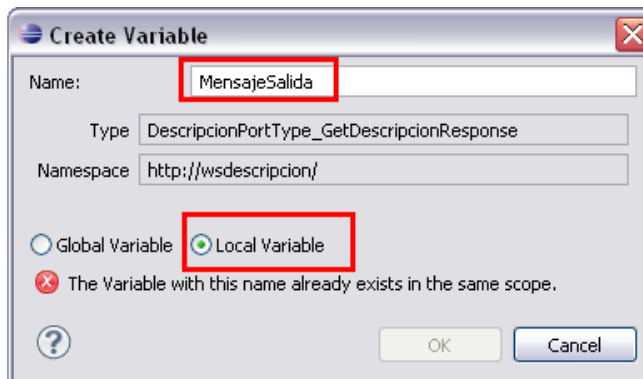
- la variable de entrada, en este caso MensajeEntrada



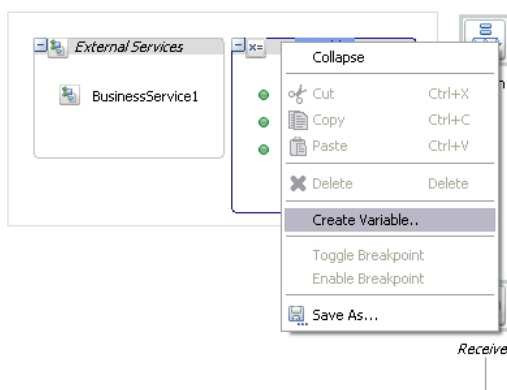
- creamos la variable de salida

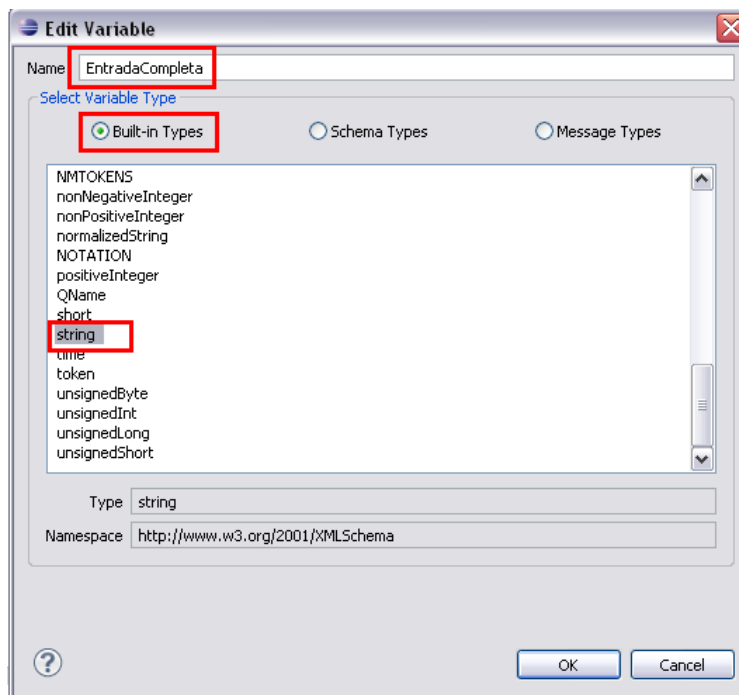


Introducimos un nombre, en este caso "MensajeSalida" y seleccionamos variable local.

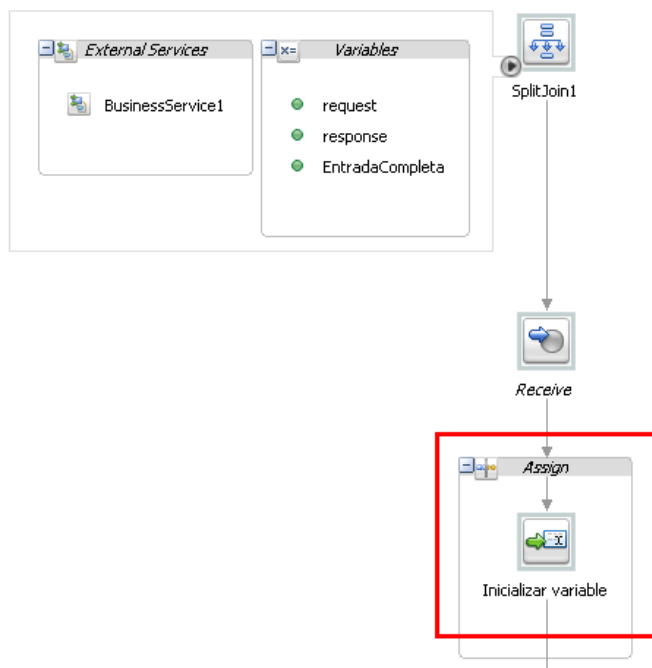


A continuación vamos a concatenar las salidas de esta invocación para devolverlas al Proxy. Para ello creamos una variable global llamada "EntradaCompleta" de tipo String.

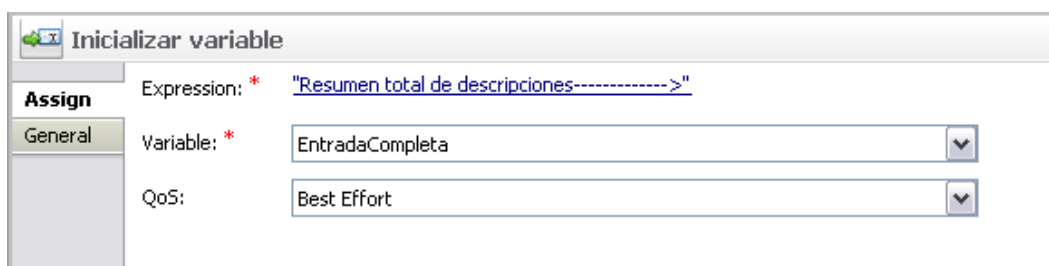




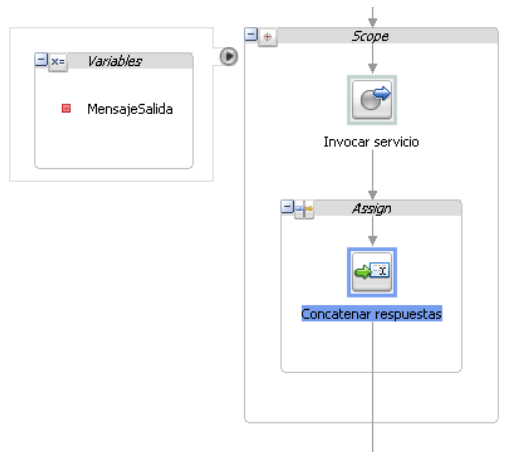
Antes del for-each añadimos un nuevo “assign” para inicializar dicha variable.



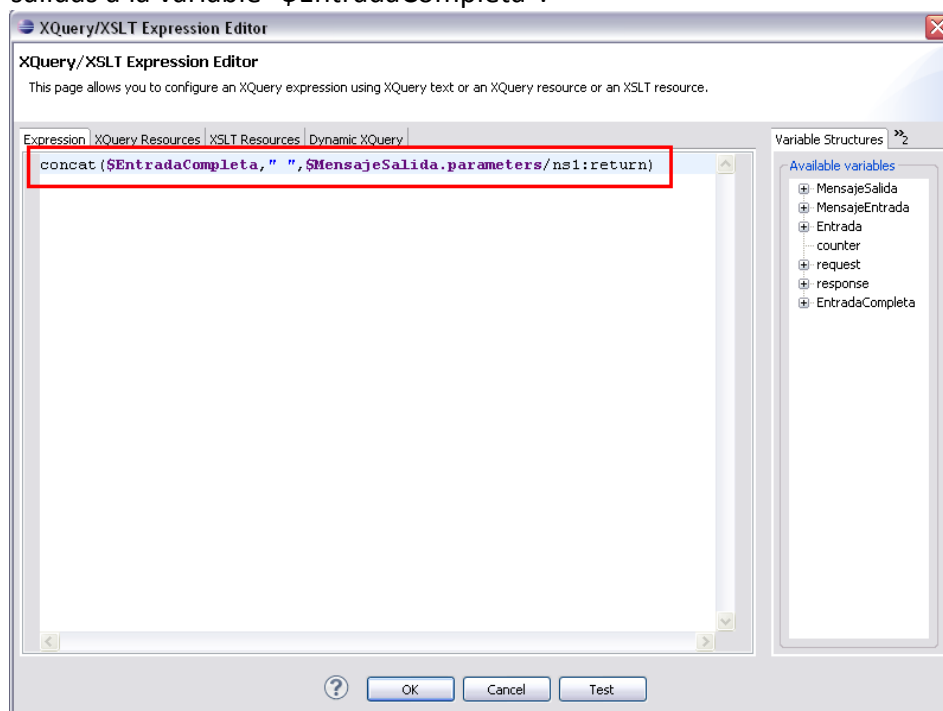
Como es un String lo inicializamos por ejemplo:



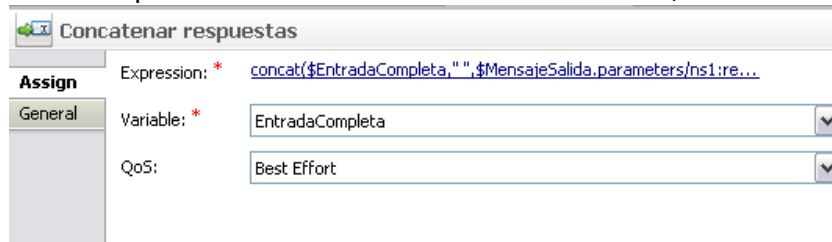
Añadimos un nuevo “assign” tras la invocación para concatenar las salidas.



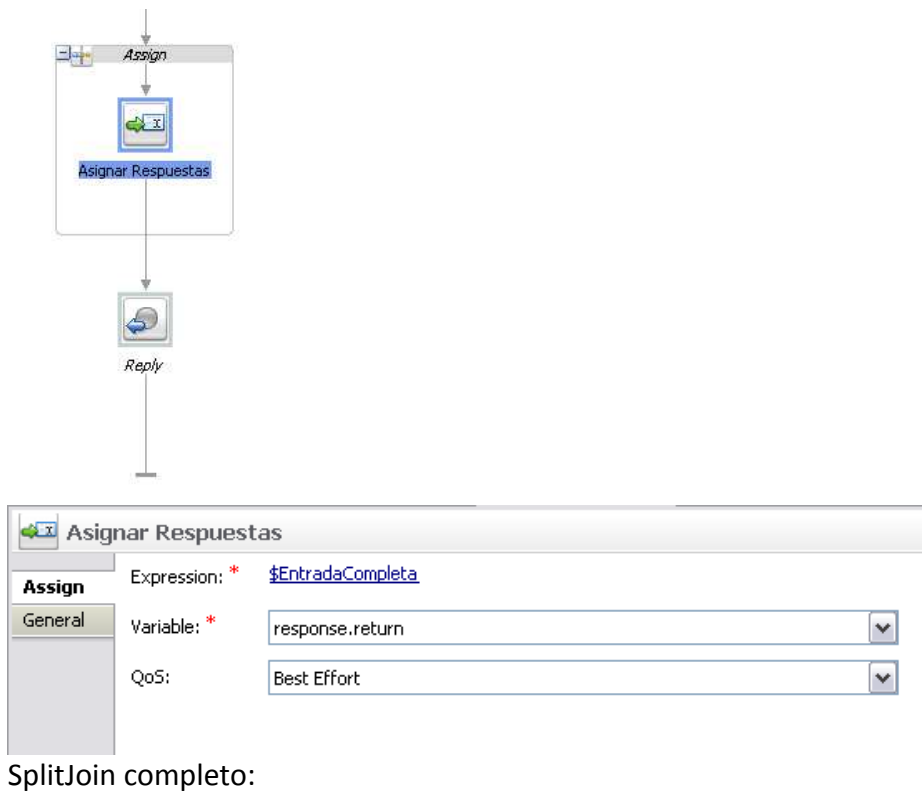
- En el campo “Expression” de las propiedades concatenamos cada una de las salidas a la variable “\$EntradaCompleta”.

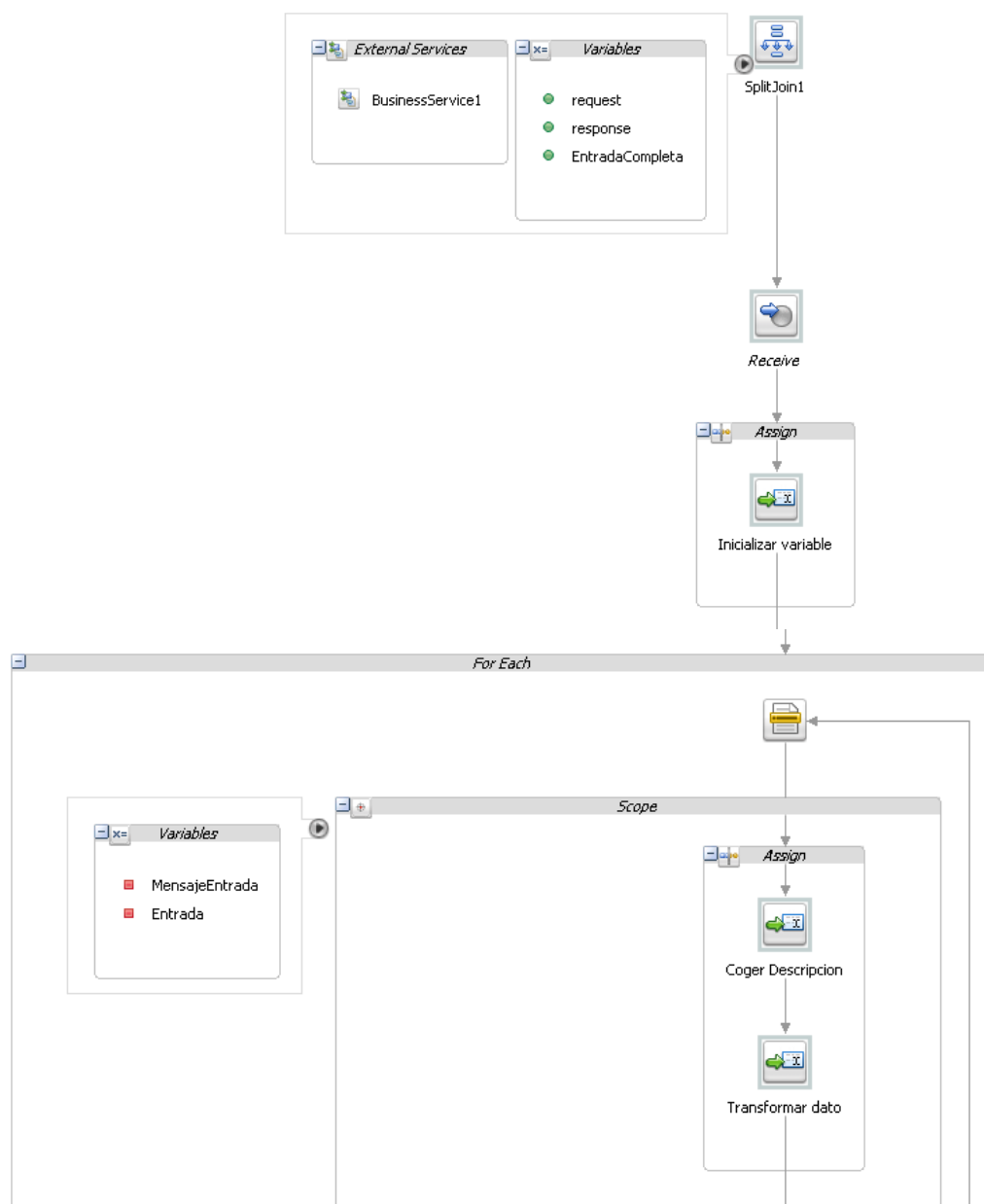


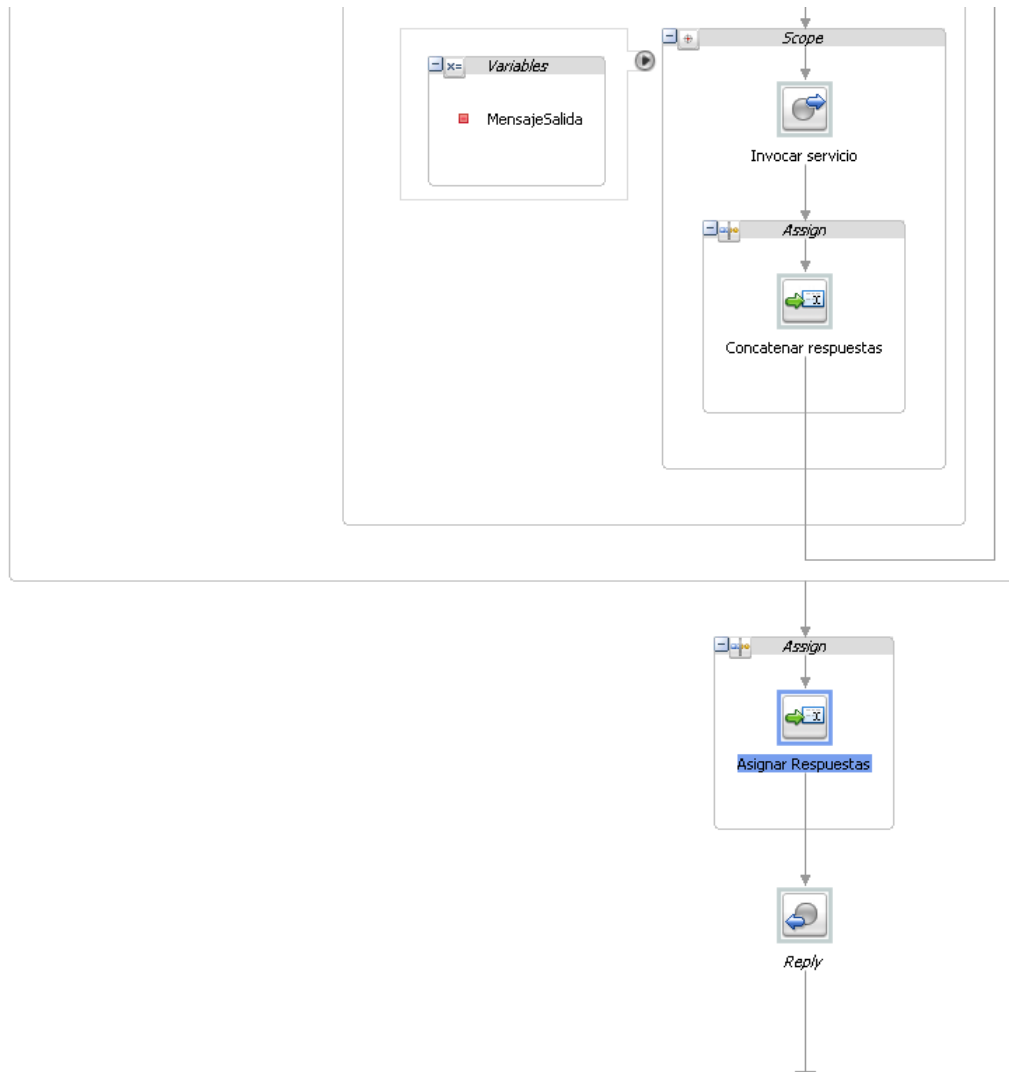
- En el campo “Variable” seleccionamos la variable “\$EntradaCompleta”.



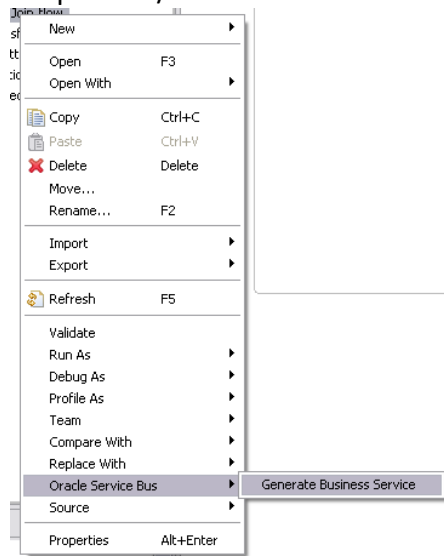
Fuera del for-each y antes del “Reply” añadimos un nuevo “Assign” en el que asignamos la variable EntradaCompleta a la variable response.



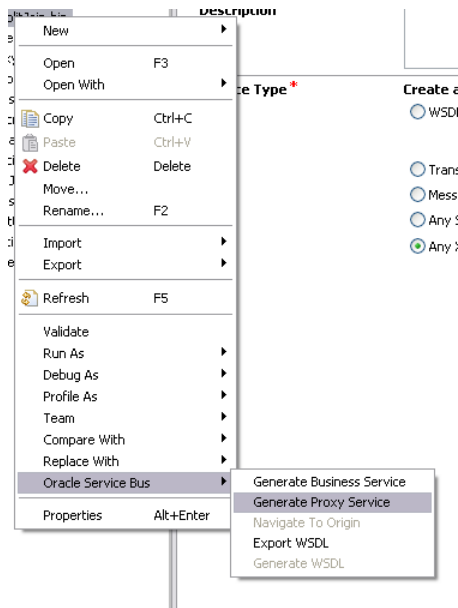




A continuación creamos el correspondiente servicio de negocio. Botón derecho sobre el SplitJoin / Generate *Business Service*:

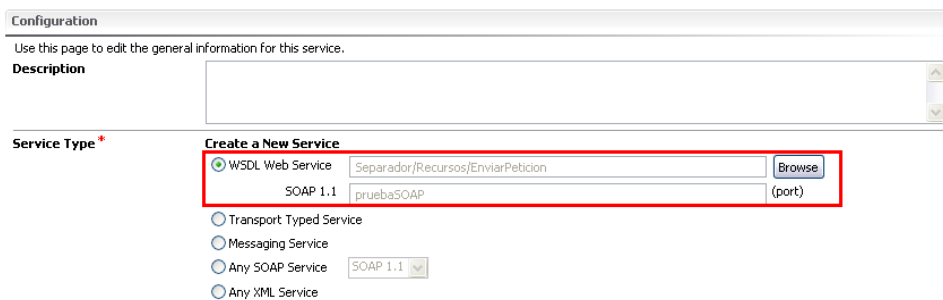


Le damos un nombre (BSSplitJoin) y seleccionamos la carpeta *Business Services*. Botón derecho sobre el Business Services recién creado / Generate *Proxy Service*.



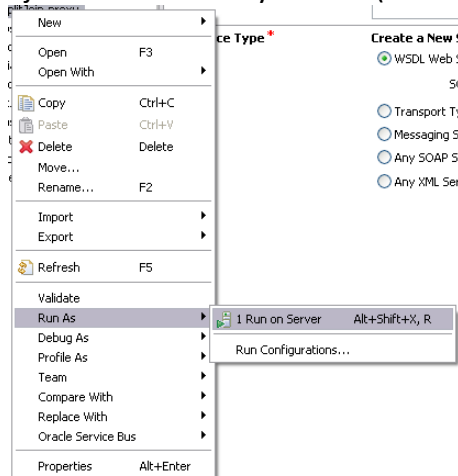
Creamos el *Proxy Service* a partir del wsdl EnviarPeticon.wsdl.

General Configuration



Prueba

Ejecutamos el Proxy creado (botón derecho sobre el Proxy Run As/Run on server:



Introducimos en total dos peticiones, es decir, dos item:

Documento de Petición

Pantalla XML

Cabecera SOAP:

```
<soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
</soap:Header>
```

*** Carga Útil:**

Examinar...

```
<exam:PurchaseOrder id="100" p="string" xsi="string" schemaLocation="string"
xmlns:exam="http://www.example.org">
  <exam:items>
    <!-- 1 or more repetitions:-->
    <exam:item id="1">
      <exam:description>primera peticion</exam:description>
      <exam:department>string</exam:department>
      <exam:price>1.5E2</exam:price>
      <exam:quantity>100</exam:quantity>
    </exam:item>

    <exam:item id="2">
      <exam:description>segunda peticion</exam:description>
      <exam:department>string</exam:department>
      <exam:price>1.5E2</exam:price>
      <exam:quantity>100</exam:quantity>
    </exam:item>
  </exam:items>
</exam:PurchaseOrder>
```

Vemos que de cada petición se muestra el campo descripción:

Caso de Uso 6: Escucha en llamada síncrona

PASO 1: desplegamos en el bus un servicio de negocio y el correspondiente servicio proxy de conexión directa. Una aplicación cliente llama al proxy normalmente.

Ejemplo:



Lo primero creamos el proyecto llamado EscuchaEnLlamadaSincronaPaso1. Creamos tres carpetas: *Business Services*, *Proxy Services* y Recursos.

En la carpeta Recursos añadimos el siguiente WSDL:

<http://a.b.c.d/GestorSoportes.asmx?wsdl> (gestor soportes)

Seleccionar Tipo de Recurso

- Servicio**
 - Servicio de Proxy
 - Servicio de Negocio
 - División-Unión
- Interfaz**
 - WSDL
 - Esquema XML
 - Política de Servicios Web
 - Enlace de JCA
- Transformación**
 - XQuery
 - XSLT
 - Archivo MFL
- Seguridad**
 - Cuenta de Servicio
 - Proveedor de Claves de Servicio
- Utilidad**
 - JAR
 - Destino de Alerta
 - Documento XML
- Bloque**
 - Recursos de URL
 - Recursos Comprimidos

Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*:

Nombre del Recurso*:

Tipo de Recurso*:

En la carpeta *Business Services* creamos el servicio de negocio asociado al anterior WSDL:

Crear un Servicio de Negocio (EscuchaEnLlamadaSincronaPaso1/Business Services/)

Configuración General

Nombre del Servicio* Servicio Gestor de Soportes

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

EscuchaEnLlamadaSincronaPaso1/Resou GestorSoportesSoap (puerto) Examinar...

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio Examinar...

☐ Servicio de Proxy Examinar...

Siguiente >> Último >> Cancelar

Pulsamos “Ultimo” y “Guardar”.

A continuación, creamos el Proxy a partir del *Business Service* que acabamos de crear.

Crear un Servicio de Proxy (EscuchaEnLlamadaSincronaPaso1/Proxy Services/)

Configuración General

Nombre del Servicio* Servicio Gestor de Soportes (proxy)

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL Examinar...

(puerto o enlace)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☒ Servicio de Negocio EscuchaEnLlamadaSincronaPaso1/Busine Examinar...

☐ Servicio de Proxy Examinar...

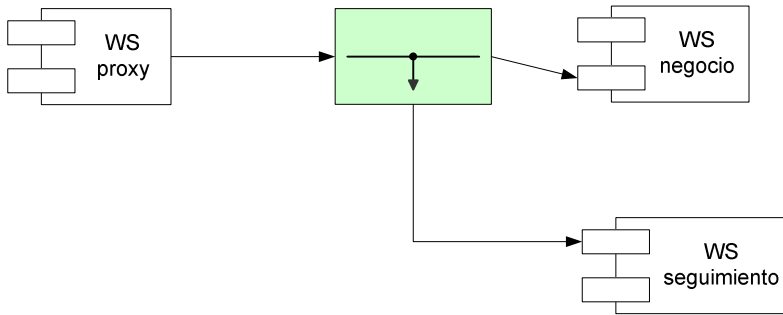
Siguiente >> Último >> Cancelar

Probamos la llamada directa al servicio de negocio.

Recursos

► **Crear Recurso:** Seleccionar Tipo de Recurso

Nombre	Tipo de Recurso	Acciones	Opciones
<input type="checkbox"/> Servicio Gestor de Soportes (proxy)	Servicio de Proxy	+ - a e b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T	



Creamos un nuevo proyecto: ServicioDeclaracionTelematica.
 Creamos tres carpetas: *Business Service*, *Proxy Service* y *Resources*.

Añadimos a la carpeta Resources los wsdl necesarios:
<http://a.b.c.d/GestorSoportes.asmx?wsdl> (gestor soportes)
<http://a.b.c.d/Seguimiento.asmx?wsdl> (seguimiento)

“Importar”.

Cargar Recursos

Revisar Recursos Cargados

Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
http://...GestorSop...	GestorSoportes	EscuchaEnLlamadaSincronaPaso2/Resources	WSDL

Elementos 1 - 1 de 1

<< Anterior Importar Cancelar

Hacemos lo mismo con el WSDL seguimiento.

En la carpeta *Business Service* creamos los servicios de negocios asociados a los anteriores WSDL:

Configuración General

Nombre del Servicio* Servicio de seguimiento

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

EscuchaEnLlamadaSincronaPaso2/Resou
SeguimientoSoap (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Siguiente >> Último >> Cancelar

“Último” y “Guardar”.

Hacemos lo mismo pero con el WSDL gestor de soporte.

Por último creamos los *Proxy Service*.

1. *Proxy Service* Servicio de seguimiento(proxy)

Creamos el Proxy a partir del WSDL Seguimiento:

Crear un Servicio de Proxy (ServicioDeclaracionTelematica/SPRINT 1/Proxy Services/)

Configuración General

Nombre del Servicio* Servicio de Seguimiento (proxy)

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

hatica/SPRINT 1/Resources/Seguimiento
SeguimientoSoap (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

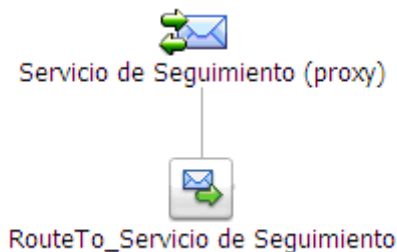
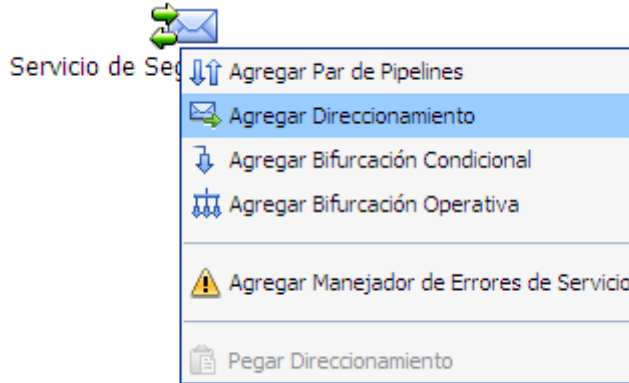
Siguiente >> Último >> Cancelar

“Último” y “Guardar”.

Entramos en el flujo de mensajes del Proxy creado:

Recursos			
Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 2 de 2			
Nombre	Tipo de Recurso	Acciones	Opciones
 Servicio de Seguimiento (proxy)	Servicio de Proxy	     	añe   


Pinchamos sobre el Proxy y añadimos direccionamiento:



Pinchamos sobre el direccionamiento creado y “Editar direccionamiento”.



“Agregar una acción” y “Comunicación/Direccionamiento”. Añadimos el servicio de negocio al cual queremos direccionar y a la operación de dicho negocio a la que queremos llamar.

En el campo Acciones de Petición añadimos una nueva acción “Validar” para comprobar que el cuerpo del mensaje cumple con un contrato expresado con un WSDL, en este caso el WSDL seguimiento.

 **Direccionar a Servicio de Seguimiento*** llamando a ApuntarResultadoLote

☐ Utilice la operación entrante para la saliente


Acciones de Petición:

 **Validar**  en variable body con respecto a **Seguimiento*** (Elemento = "ApuntarResultadoLote")

☐ Guardar resultado de la validación en variable *

☒ Emitir error en caso de fallo de validación

Acciones de Respuesta:

 **Agregar una Acción**

“Guardar todo”.

2. *Proxy Service* Servicio Gestor de Soportes
Creamos el Proxy a partir del WSDL Seguimiento:

Crear un Servicio de Proxy (ServicioDeclaracionTelematica/SPRINT 1/Proxy Services/)

Configuración General

Nombre del Servicio* Servicio Gestor de Soportes (proxy)

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

ca/SPRINT 1/Resources/GestorSoportes GestorSoportesSoap (puerto) Examinar...

SOAP 1.1

Examinar...

Examinar...

Examinar...

Siguiente >> Último >> Cancelar

“Último” y “Guardar”.

Entramos en el flujo de mensajes del Proxy creado:

Recursos

► Crear Recurso: Seleccionar Tipo de Recurso

Nombre	Tipo de Recurso	Acciones	Opciones
Servicio de Seguimiento (proxy)	Servicio de Proxy		añe
Servicio Gestor de Soportes (proxy)	Servicio de Proxy		añe

Elementos 1 - 2 de 2

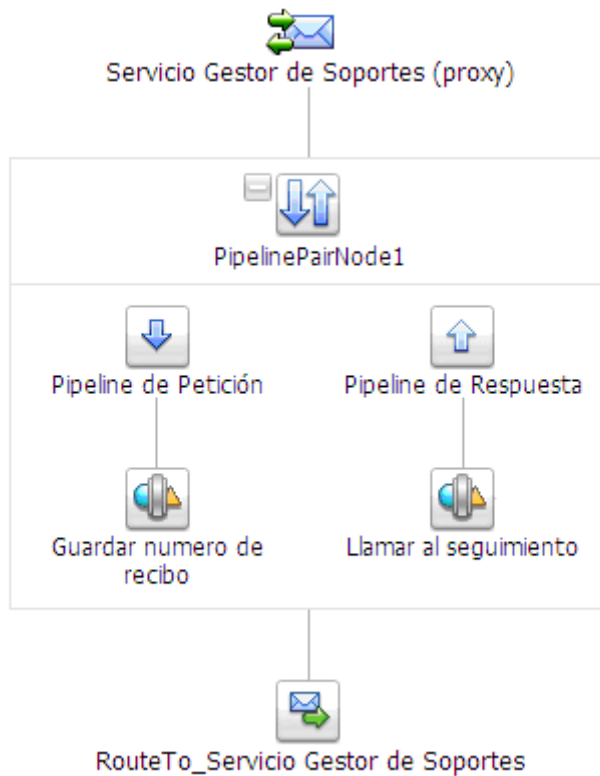
Suprimir

Pinchamos sobre el Proxy creado “Agregar Par de Pipelines”.

Pinchamos sobre el Pipeline de petición y “agregar etapa”.

Pinchamos sobre el Pipeline de Respuesta y “agregar etapa”.

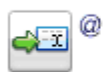
Pinchamos sobre PipelinePairNode1 y “Agregar Direccionamiento”.



Editamos Guardar número de recibo.

Añadimos acción “Asignar”.

Pinchamos en expresión y seleccionamos la variable numeroRecibo del body. Asignamos el resultado a la variable numeroRecibo:



Asignar `$body/ser:Valid...` a variable `numeroRecibo`

Editamos direccionamiento RouteTo_Servicio Gestor de Soportes.

Añadimos “Comunicación/Direccionamiento”. Seleccionamos el servicio de negocio Servicio Gestor de Soportes.

Direccionar a Servicio Gestor de Soportes* llamando a **Operación**

☒ Utilice la operación entrante para la saliente

Acciones de Petición:

- Agregar una Acción

Acciones de Respuesta:

- Agregar una Acción

Editamos la etapa Llamar al seguimiento.

Añadimos acción “Asignar”.

En expresión cogemos la variable ResultadoValidacion del cuerpo de la respuesta y lo guardamos en la variable ResultadoValidacion.

ORACLE® Service Bus 11gR1

Estructuras de Variables

body

- \$body - ValidarLote (request)
- ValidarLote
- \$body - ValidarLote (response)
- ValidarLoteResponse
 - ResultadoValidacion**
 - ResultadoCorrecto
 - ReglasRotas
 - Reg0

Editor de Expresiones XQuery/XSLT : Pipeline de Respuesta: Llamar al seguimiento

Guardar Validar Probar Cancelar Limpiar Guardar Todo

Texto de XQuery | Recursos de XQuery | Recursos de XSLT | XQuery Dinámica

`$body/ser:ValidarLoteResponse/ges:ResultadoValidacion`

Inspector de Propiedades

`$body/ser:ValidarLoteResponse/ges:ResultadoValidacion`

Definiciones de Espacio de Nombres

Funciones XQuery

Estructuras de Variables

“Guardar”.

Añadimos otra acción “Asignar”. En el texto ponemos lo siguiente:

```
<ser:ApuntarResultadoLote xmlns:ser="http://a.b.c.d" xmlns:ges="xxxxxx">
  <ser:numeroLote>{numeroRecibo/text()}</ser:numeroLote>
</ser:ApuntarResultadoLote>
```

Esto es la entrada esperada por el servicio de negocio Seguimiento. En el campo numeroLote añadimos el campo numeroRecibo del mensaje de entrada. Lo guardamos en la variable ApuntarResultadoLote.

ORACLE® Service Bus 11gR1

Definiciones de Espacio de Nombres

Espacios de Nombres por Defecto

bpel-10g	http://www.bea.com/wli/sb/transport/bpel10g
ctx	http://www.bea.com/wli/sb/context
dsp	http://www.bea.com/dsp/transport/sb
ejb	http://www.bea.com/wli/sb/transport/ejb
email	http://www.bea.com/wli/sb/transport/email
file	http://www.bea.com/wli/sb/transport/file

Espacios de Nombres de Variables

ges	http://gestorsoporte.sdt.navarra.es
ser	http://servicios.sdt.navarra.es/

Espacios de Nombres Definidos por el Usuario [Agregar Espacio de Nombres](#)

No se ha encontrado ningún espacio de nombres.

Definiciones de Espacio de Nombres

Funciones XQuery

Estructuras de Variables

Bienvenido, weblogic Conectado a: poc_osbsoa Inicio Consola de Oracle WLS Desconectar Ayuda

Sesión de weblogic Creado el 9/12/10 1:

Editor de Expresiones XQuery/XSLT : Pipeline de Respuesta: Llamar al seguimiento

Guardar Validar Probar Cancelar Limpiar

Texto de XQuery Recursos de XQuery Recursos de XSLT XQuery Dinámica

```
<ser:ApuntarResultadoLote xmlns:ser="http://..." xmlns:ges="http://...">
  <ser:numeroLote>{ser:numeroLote}</ser:numeroLote>
  {ser:ResultadoValidacion}
</ser:ApuntarResultadoLote>
```

Inspector de Propiedades

“Guardar todo”.

Añadimos “Llamada a Servicio”. Llamamos al Proxy Servicio de Seguimiento con la operación ApuntarResultadoLote. Como variable de entrada añadimos la variable ApuntarResultadoLote creada anteriormente y la salida la guardamos en la variable ResuApuntarResultadoLote.

Asignar \$body/ser:Valid... a variable ResultadoValidacion *

Asignar <ser:ApuntarRes... a variable ApuntarResultadoLote *

Llamada de Servicio a Servicio de Seguimiento (proxy) llamando a ApuntarResultadoLote *

☐ Configurar Cuerpo SOAP ☒ Configurar Documento de Carga Útil

Variable del Documento de Petición: ApuntarResultadoLote *

Variable del Documento de Respuesta: ResuApuntarResultadoLote *

Cabecera de la Petición SOAP:

Cabecera de la Respuesta de SOAP:

Acciones de Petición:

☒ Agregar una Acción

Acciones de Respuesta:


☒ Agregar una Acción

A continuación vamos a gestionar los errores, es decir que capture los errores y lance alertas.

Para ello es necesario crear un destino de alerta.

Lo primero creamos un servidor SMTP:



 **Editar Servidor SMTP:** (...)

Nombre*	(...)
Descripción	<input type="text"/>
URL del Servidor*	<input type="text"/>
Número de Puerto*	<input type="text" value="25"/>
Usuario	<input type="text"/>
Nueva Contraseña	<input type="text"/>
Confirmar Contraseña	<input type="text"/>
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Una vez creado el servidor, procedemos a crear el destino de la alerta dentro de la carpeta Resources.

Referencias

Referencias de otros recursos

Carpetas

► Introducir Nombre

Nombre

Suprimir

Recursos

► Crear Recurso:

Seleccionar Tipo de Recurso

Servicio

- Servicio de Proxy
- Servicio de Negocio
- División-Unión

Interfaz

- WSDL
- Esquema XML
- Política de Servicios Web
- Enlace de JCA

Transformación

- XQuery
- XSLT
- Archivo MFL

Seguridad

- Cuenta de Servicio
- Proveedor de Claves de Servicio

Utilidad

- JAR
- Destino de Alerta
- Documento XML

Bloque

- Recursos de URL
- Recursos Comprimidos

Seleccionar Tipo de Recurso

Nombre del Recurso*	Lista de correo de operación y soporte
Descripción del Recurso	
Interrupción SNMP	<input checked="" type="radio"/> Sí <input type="radio"/> No
Informes	<input type="radio"/> Sí <input checked="" type="radio"/> No
Registro de Alertas	<input checked="" type="radio"/> Sí <input type="radio"/> No
Destinatarios del Correo Electrónico	<p>Destinatarios</p> <p><input type="text"/></p> <p>Agregar</p>
Destinos de JMS	<p>Destinos</p> <p>No hay ningún destino de JMS que mostrar.</p> <p>Agregar</p>

Guardar Cancelar

Añadimos el destinatario de correo electrónico. Para ello introducimos una dirección de correo electrónico y el servidor SMTP que acabamos de crear.

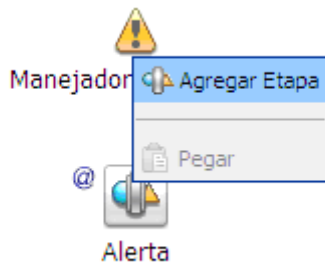
Destinatarios del Correo*	<p>El formato de destinatarios de correo es usuario1@host[,usuario2@host]</p> <p>fernando.izco.garcia@navarra.es</p>
Servidor SMTP	CENTEXC01
Sesión de Correo	Ninguno Disponible
Nombre del Remitente	
Dirección de Origen	
Nombre de Respuesta	
Dirección de Respuesta	
Tiempo de Conexión	0
Codificación de Solicitud	iso-8859-1

Guardar Cancelar

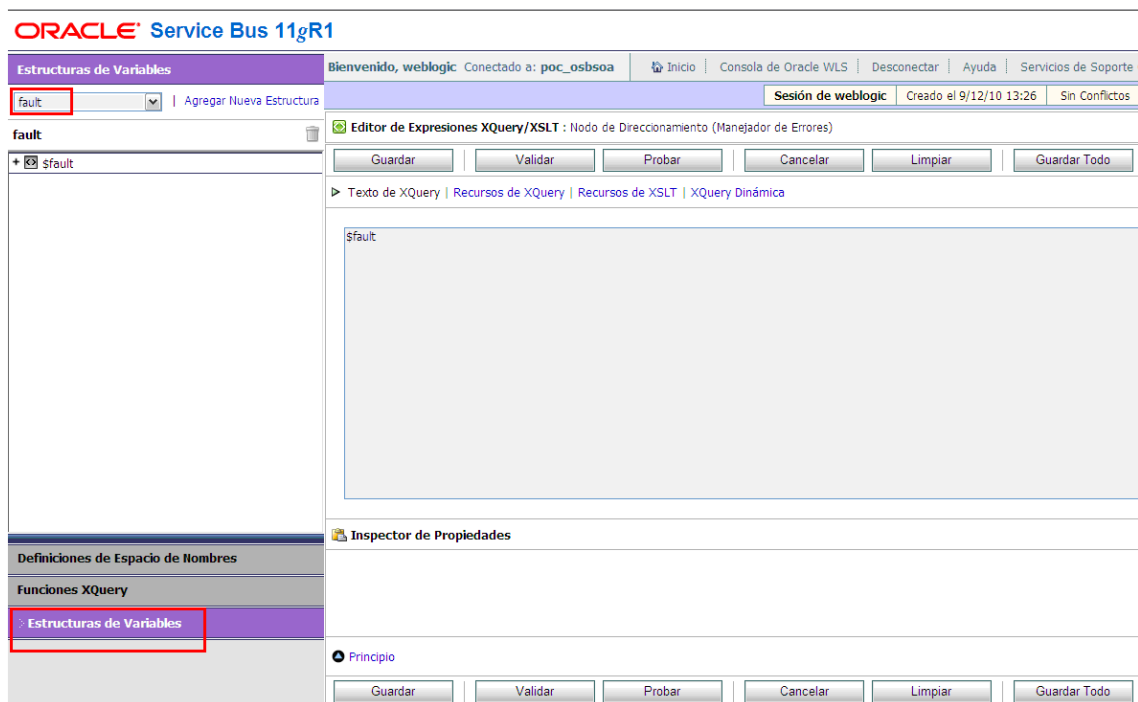
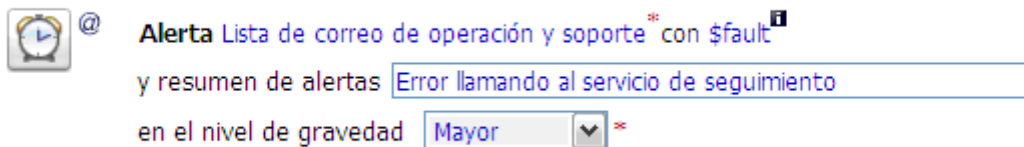
Una vez creado el destino de alerta, vamos al flujo de mensaje:

1. Proxy seguimiento

Pinchamos en el direccionamiento y “Agregar Manejador de Errores de Direccionamiento”.
“Agregar Etapa”.



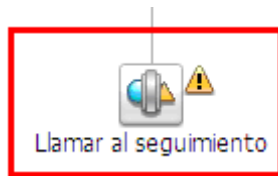
Añadimos una acción “Alerta”. Seleccionamos el destino de alerta que acabamos de crear y el error que se encuentra en la variable \$fault.



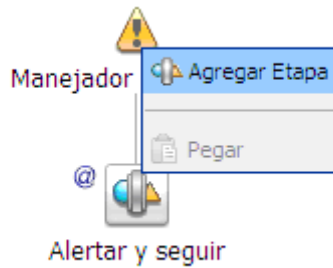
2. Proxy Gestor de soporte

El gestor de errores de la llamada al servicio de seguimiento, además de lanzar la alerta, reanuda la ejecución, es decir, se ‘come’ el error.

Pinchamos en la etapa Llamar al seguimiento y “Agregar Manejador de Errores de Etapa”.



“Agregar Etapa”.



Añadimos una acción “Alerta”. Seleccionamos el destino de alerta que acabamos de crear y el error que se encuentra en la variable \$fault.

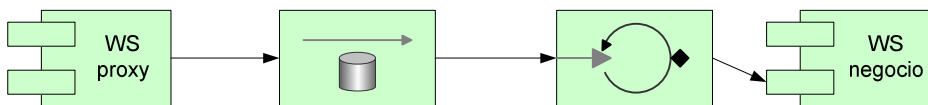
Añadimos una acción “Reanudar” para que si falla continuemos con la ejecución.

Caso de Uso 7: Canal de entrega garantizada

Alcance:

El objetivo es demostrar la implementación de un canal de mensajes de entrega garantizada, que asegura la entrega de los mensajes aunque el sistema de mensajería falle.

Entre el servicio proxy y el de negocio tendremos un canal de mensajes de entrega garantizada. El servicio de negocio será especialmente lento en el tratamiento de los mensajes.



La aplicación cliente llamará al proxy entregando un determinado número de mensajes. Como el consumo de mensajes por parte del servicio de negocio es lento, nos dará tiempo a parar el bus o el servidor de aplicaciones. Después de reiniciarlo, se consumen los mensajes pendientes. Al cabo de un tiempo todos los mensajes han llegado al servicio de negocio.

URL WSLento60segundosService:

<http://a.b.c.d:8081/WSLento60segundos-WSLento60segundos-context-root/WSLento60segundosPort?WSDL>

Se ha desarrollado un *Web Service* que tarda en contestar 60 segundos.

Creamos un nuevo proyecto llamado CanalEntregaGarantizada. Añadimos tres carpetas: *Proxy Services*, *Business Services* y Recursos.

En la carpeta Recursos añadimos el WSLento60segundosService.



Añadimos la URL del recurso:

Cargar Recursos

Cargar Recursos desde URL

URL/Ruta de Acceso*

Nombre del Recurso*

Tipo de Recurso*

“Siguiente”.

Cargar Recursos

Revisar Recursos Cargados

Elementos 1 - 2 de 2

<input checked="" type="checkbox"/>	Nombre de Archivo	Nombre	Ruta de Acceso	Tipo de Recurso
<input checked="" type="checkbox"/>	http://10.222.2.2:8080/WSLento60segundos...	WSLento60segundosPort	CanalEntregaGarantizada/Recursos	WSDL
<input checked="" type="checkbox"/>	http://10.222.2.2:8080/WSLento60segundos...	XMLSchema_1211130486	CanalEntregaGarantizada/Recursos	Esquema XML

Elementos 1 - 2 de 2

“Importar”.

Añadimos un nuevo servicio de negocio en la carpeta *Business Services*:

CanalEntregaGarantizada/Business Services

Seleccionar Tipo de Recurso

Servicio

Servicio de Proxy

Servicio de Negocio

División-Unión

Interfaz

WSDL

Esquema XML

Política de Servicios Web

Enlace de JCA

Transformación

XQuery

XSLT

Archivo MFL

Seguridad

Cuenta de Servicio

Proveedor de Claves de Servicio

Utilidad

JAR

Destino de Alerta

Documento XML

Bloque

Recursos de URL

Recursos Comprimidos

Crear Recurso:

Ponemos un nombre y seleccionamos el wsdl que acabamos de crear en la carpeta recursos. “Siguiente”.

Crear un Servicio de Negocio (CanalEntregaGarantizada/Business Services/)

Configuración General

Nombre del Servicio*

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL
 (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Seleccionamos el protocolo http (está por defecto) y añadimos una URI de punto final. Dejamos el resto de campos como están y pulsamos “último”.

Crear un Servicio de Negocio (CanalEntregaGarantizada/Business Services/BusinessService1)

Configuración de Transporte

Protocolo*

Algoritmo de Equilibrio de Carga

URI de Punto Final*

Formato:

URI EXISTENTES	OPCIONES
<input type="text" value="http://*"/> <input type="text" value="/WSLento60segundos-WSLento60segundos-context-root/WSLento60segundosPort"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>

Recuento de Reintentos

Intervalo de Iteración de Reintentos

Reintentar tras Errores de Aplicación ☒ Sí ☐ No

Guardar.

Vamos al servicio de negocio que acabamos de crear:

Recursos

► Crear Recurso:

Elementos 1 - 1 de 1

Nombre	Tipo de Recurso	Acciones	Opciones
<input type="checkbox"/> <input type="text" value="BusinessService1"/>	Servicio de Negocio	<input type="button" value="✎"/> <input type="button" value="✖"/>	añe <input type="button" value="👤"/> <input type="button" value="👤"/> <input type="button" value="👤"/>

Elementos 1 - 1 de 1

Pestaña Configuración Operativa.

Marcamos estado de regulación.

Simultaneidad máxima: Introducimos “1”, es decir, el número de mensajes tratados a la vez por cada URI.

Cola de regulación: Introducimos “1000” por ejemplo, es decir, el número máximo de mensajes en espera que se guardan en caso de que la URI se encuentren ocupadas. Si supera este valor, los mensajes se envían a la cola de error.

Detalles de Configuración **Configuración Operativa** Reglas de Alertas de SLA Políticas

Configuración General

Estado ☒ Activado

URI de Punto Final Fuera de Línea ☐ Activar con Intervalo de Reintentos 0 horas 0 mins. 0 segs.

Supervisión

Supervisión ☐ Activado

Intervalo de Agregación 0 horas 10 mins.

Alertas de SLA ☒ Activar Alertas en nivel Normal o superior

Regulación

Estado de Regulación ☒ Activado

Simultaneidad Máxima* 1

Cola de Regulación* 1000 mensajes

Caducidad del Mensaje 0 milisegundos

Rastreo

Rastreo de Mensajes ☐ Activado

Nivel de Detalle Terse

Límite de Rastreo de Carga Útil 8 Kilobytes

Codificación por Defecto

Almacenamiento en Caché de Resultados

Estado de Caché de Resultados ☒ Activado

Atrás Actualizar Restablecer

“Actualizar”.

A continuación creamos el PS2 a partir del BusinessService1

Seleccionar Tipo de Recurso

Servicio

- Servicio de Proxy
- Servicio de Negocio
- División-Unión

Interfaz

- WSDL
- Esquema XML
- Política de Servicios Web
- Enlace de JCA

Transformación

- XQuery
- XSLT
- Archivo MFL

Seguridad

- Cuenta de Servicio
- Proveedor de Claves de Servicio

Utilidad

- JAR
- Destino de Alerta
- Documento XML

Bloque

- Recursos de URL
- Recursos Comprimidos

Crear Recurso: Seleccionar Tipo de Recurso

Le damos un nombre: PS2 y lo creamos a partir de cualquier servicio SOAP.
“Siguiente”.

Crear un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/)

Configuración General	
Nombre del Servicio*	PS2
Descripción	
Tipo de Servicio*	<p>Crear un Servicio Nuevo</p> <p><input type="radio"/> Servicio Web WSDL <input type="button" value="Examinar..."/></p> <p><input type="radio"/> Servicio de Transporte según Configuración</p> <p><input type="radio"/> Servicio de Mensajes</p> <p><input checked="" type="radio"/> Cualquier Servicio SOAP SOAP 1.1 <input type="button" value="Examinar..."/></p> <p><input type="radio"/> Cualquier Servicio XML</p> <p>Crear a partir de Servicio Existente</p> <p><input type="radio"/> Servicio de Negocio <input type="button" value="Examinar..."/></p> <p><input type="radio"/> Servicio de Proxy <input type="button" value="Examinar..."/></p>
<input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

Le indicamos que utilice el protocolo de transporte JMS. La URI de punto final se crea de manera automática haciendo referencia a la cola que va a utilizar. Las colas se crean de manera automática una vez que ejecutemos el Proxy.

Crear un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/PS2)

Configuración de Transporte	
Protocolo*	jms
URI de Punto Final*	Formato: <code>jms:///((host:port)),((host:port))*(((host:port)?)/FactoryJndiName/QueueJndiName)</code> <code>jms://10.10.10.10:10101/weblogic.jms.XAConnectionFactory/PS2LentoPetición</code>
Obtener Todas las Cabeceras	<input checked="" type="radio"/> Sí <input type="radio"/> No
<input type="button" value="Anterior <<"/> <input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

“Siguiente”.

Seleccionamos la siguiente configuración de transporte:

Editar un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/PS2)	
Configuración de Transporte de JMS	
Tipo de Destino	<input checked="" type="radio"/> Cola <input type="radio"/> Tema
Se Necesita Respuesta	<input type="checkbox"/>
Política de Distribución	default
Codificación de Peticiones	UTF-8
Cuenta de Servicio de JMS	<input type="button" value="Examinar..."/>
Configuración Avanzada	
<input type="button" value="Anterior <<"/> <input type="button" value="Siguiente >>"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

Si lo que queremos es tener un canal de entrega garantizada no debemos marcar “Se necesita respuesta” ya que si enviamos un número de mensajes y paramos el servidor, al arrancarlo se forma un bucle de mensajes en los que se intenta hacer rollback de la transacción pero no se consigue por lo que se forma un bucle infinito.

Desplegamos Configuración avanzada.

Introducimos el número de reintentos en caso de fallo, el tiempo entre cada uno de esos reintentos (en milisegundos), el destino de error (en este caso la cola ColaErrorLento60segundos) y en caso de que después de todos los reintentos siga fallando se redirige a esta cola.


Configuración Avanzada	
Usar SSL	<input type="checkbox"/>
Selector de Mensajes	
Suscripción Duradera	<input type="checkbox"/>
Recuento de Reintentos	3
Intervalo de Reintentos	5000
Destino de Error	ColeErrorLento60segundos
Política de Caducidad	Redireccionar
Se necesita XA	<input type="checkbox"/>
Timeout de JNDI	0
<input type="button" value="Anterior"/> <input type="button" value="Siguiente >"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	



“Siguiente”. Seleccionamos misma transacción para respuesta. Si algo ocurre se produce rollback en la transacción.

Editar un Servicio de Proxy (CanalPuntoAPuntoConfiablePaso2/Proxy Services/PS2)	
Manejo de Mensajes	
Transacción Necesaria	<input type="checkbox"/> Activado
Misma Transacción para Respuesta	<input checked="" type="checkbox"/> Activado
Flujo de Contenido	<input type="checkbox"/> Activado Tipo de Buffer <input type="radio"/> Buffer de Memoria <input type="radio"/> Buffer de Disco Compresión <input type="checkbox"/> Activado
<input type="button" value="Anterior"/> <input type="button" value="Siguiente >"/> <input type="button" value="Último >>"/> <input type="button" value="Cancelar"/>	

“Último” y “Guardar”.

Entramos en el flujo de mensajes del PS2 que acabamos de crear. Añadimos “Direccionamiento” al BusinessService1. Añadimos “Opciones de direccionamiento”. Seleccionamos Calidad del Servicio “Una Sola Vez”, para evitar que se dupliquen los mensajes.


 Direccionar a BusinessService1* llamando a ServicioLento

Acciones de Petición:	
	Opciones de Direccionamiento para sustituir las siguientes configuraciones por defecto: <input type="checkbox"/> URI: <Expresión>* <input checked="" type="checkbox"/> Calidad del Servicio: Una Sola Vez* <input type="checkbox"/> Modo: Petición* <input type="checkbox"/> Intervalo de Reintentos: (segundo)* <input type="checkbox"/> Recuento de Reintentos: * <input type="checkbox"/> Prioridad: <Expresión>*
Acciones de Respuesta:	
 Agregar una Acción	

Guardar todo.

A continuación creamos el Proxy principal al que se accede por HTTP y hace referencia al Proxy anteriormente creado.

Le damos un nombre (PS1) y lo creamos a partir del WSDL. “Siguiente”.

Crear un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/)

Configuración General

Nombre del Servicio* PS1

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL
 (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

Dejamos el protocolo de transporte a HTTP. Pulsamos “siguiete” hasta el final sin hacer ningún cambio más.

Crear un Servicio de Proxy (CanalEntregaGarantizada/Proxy Services/PS1)

Configuración de Transporte

Protocolo* http

URI de Punto Final* Formato: /someName
/CanalEntregaGarantizada/Proxy_Services/PS1

Obtener Todas las Cabeceras

☒ Sí

☐ No

Por último vamos al flujo de mensajes del PS1.

Recursos

► Crear Recurso:

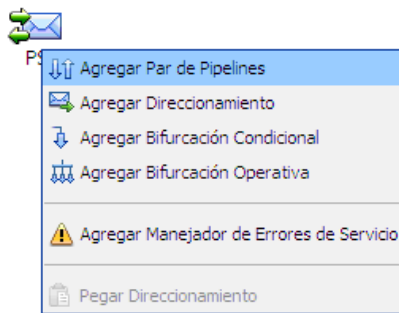
Elementos 1 - 2 de 2

Nombre	Tipo de Recurso	Acciones	Opciones
PS1	Servicio de Proxy	<input checked="" type="button" value="Publish"/> <input type="button" value="Route to"/> <input type="button" value="Delete"/>	a e <input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
PS2	Servicio de Proxy	<input type="button" value="Publish"/> <input type="button" value="Route to"/> <input type="button" value="Delete"/>	a e <input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

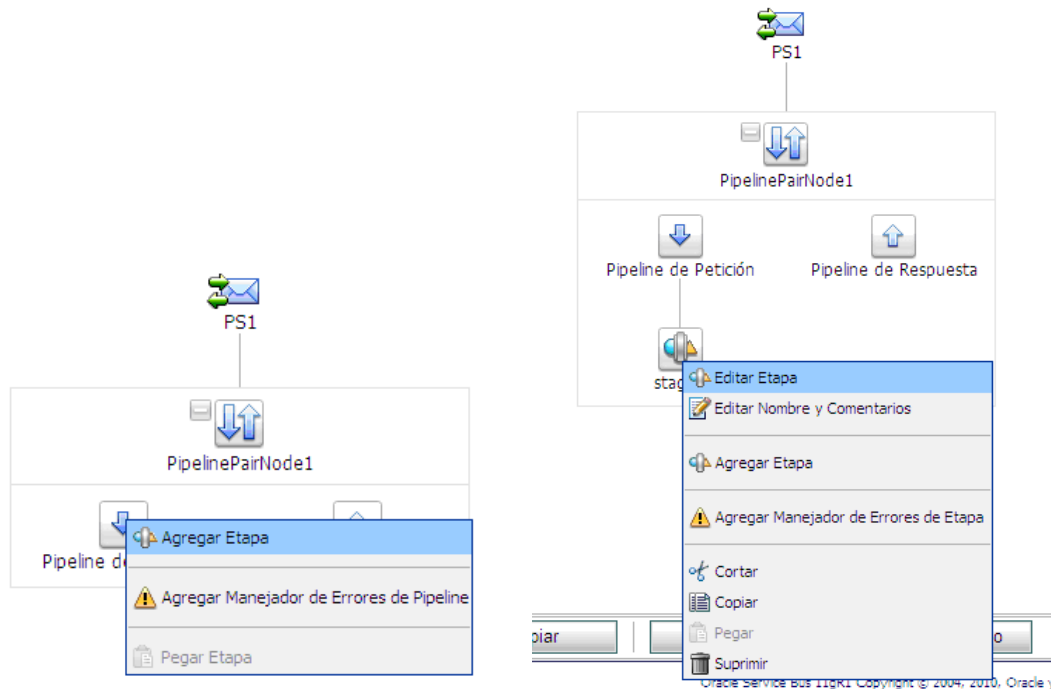
Elementos 1 - 2 de 2

Como queremos mandar el mensaje pero no queremos esperar a la respuesta sino obtener una respuesta inmediata debemos utilizar “Publish” en vez de “Route to”. Por ello:

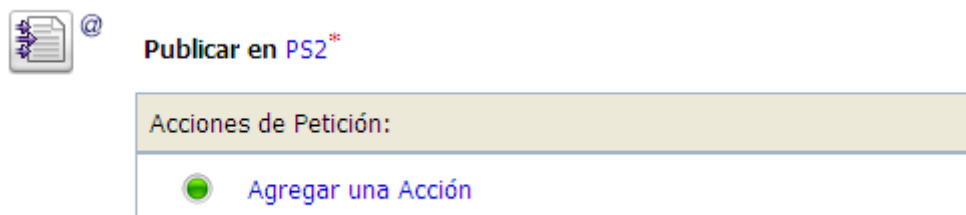
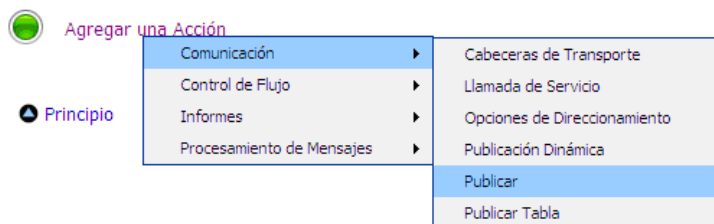
14. Añadimos un par de pipelines



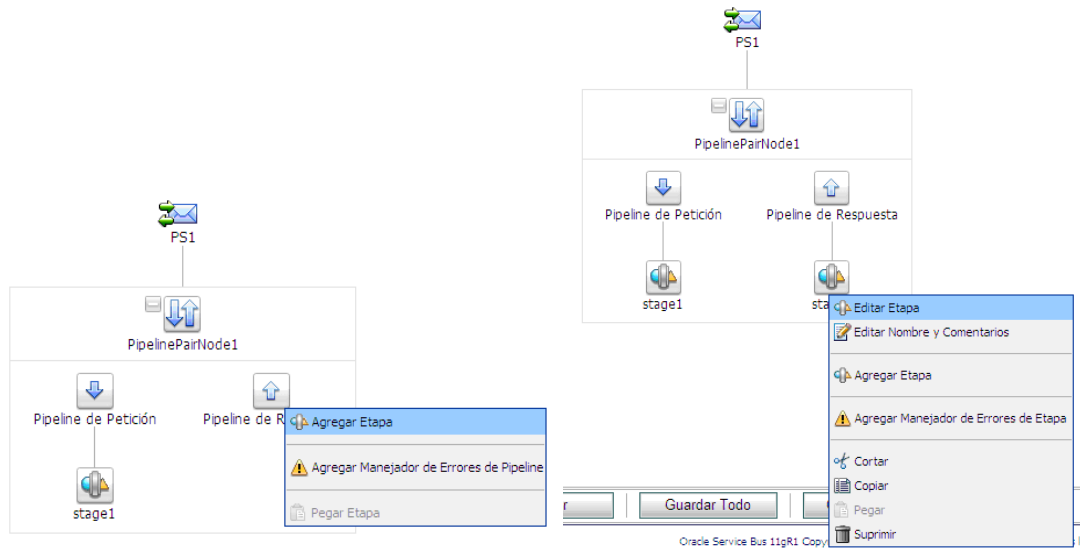
15. En el pipeline de petición añadimos una nueva etapa.




16. Añadimos una nueva acción “Publicar”. Seleccionamos al servicio al que queremos direccionar, en este caso PS2 y seleccionamos el único método que existe.




17. En el pipeline de respuesta añadimos una nueva etapa.



18. Añadimos una acción “Asignar” para crear la respuesta del Proxy.

 **Asignar** "Solicitud envi..." a variable



 **Sustituir** en variable con \$Solicitud

☐ Sustituir nodo completo

☒ Sustituir contenido del nodo

Prueba en la consola del OSB

Ejecutamos varias veces el PS1.

Recursos				
▶ Crear Recurso: <input type="text" value="Seleccionar Tipo de Recurso"/>				
Elementos 1 - 2 de 2				
<input type="checkbox"/> Nombre	Tipo de Recurso	Acciones		Opciones
<input type="checkbox"/> PS1	Servicio de Proxy			 
<input type="checkbox"/> PS2	Servicio de Proxy			 
Elementos 1 - 2 de 2				
<input type="button" value="Suprimir"/>				

Obtenemos una respuesta inmediata, mientras en segundo plano se está ejecutando la operación.

Prueba de Servicios de Proxy - PS1 Ayuda

Atrás Cerrar

Documento de Petición

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <wsd:IntroducirNombre xmlns:wsd="http://wsdlunidireccionalpocho/">
      <!--Optional:-->
      <arg0>string</arg0>
    </wsd:IntroducirNombre>
  </soapenv:Body>
</soapenv:Envelope>
```

Documento de Respuesta

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>Solicitud enviada</soapenv:Body>
</soapenv:Envelope>
```

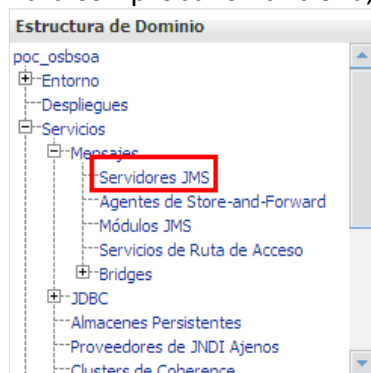
Metadatos de Respuesta

```
<con:metadata xmlns:con="http://www.bea.com/wli/sb/test/config">
  <tran:headers xsi:type="http:HttpHeaders" xmlns:http="http://www.bea.com/wli/sb/transport/http"
    xmlns:tran="http://www.bea.com/wli/sb/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <http:Content-Type>text/xml; charset=utf-8</http:Content-Type>
  </tran:headers>
  <tran:response-code xmlns:tran="http://www.bea.com/wli/sb/transport">0</tran:response-code>
  <tran:encoding xmlns:tran="http://www.bea.com/wli/sb/transport">utf-8</tran:encoding>
</con:metadata>
```

Rastreo de Llamada

- (recibiendo petición)
- PipelinePairNode1
 - stage1
- (realizando eco de petición)
- PipelinePairNode1
 - stage1

Para comprobar si funciona, vamos al weblogic. Servicios / Mensajes / Servidores JMS



Elegimos el servidor wlsbJMSServer.

Resumen de Servidores JMS

Los servidores JMS actúan como contenedores de gestión para las colas y los temas en los módulos JMS que están dirigidos a ellos.

En esta página se resumen los servidores JMS que se han creado en el dominio de WebLogic Server actual.

[Personalizar esta Tabla](#)

Servidores JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Almacén Persistente	Destino	Servidor Actual	Estado
<input type="checkbox"/>	BPMJMServer	BPMJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	JMSserver	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	SOAJMSserver	SOAJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	UMSJMSserver	UMSJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/>	wlsbJMSserver	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/>	WseeJMSserver	WseeFileStore	osb_server1	osb_server1	✓ OK

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior | Siguiente

Pestaña Supervisión / Destinos Activos.

En este ejemplo, se ha ejecutado el PS1 8 veces. Vemos que las 8 peticiones se encuentran encoladas:

Valores para wlsbJMSserver

Configuración Registro Destinos **Supervisión** Control Notas

Supervisión **Destinos Activos** Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

[Personalizar esta Tabla](#)

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

Mostrando 1 a 9 de 9 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources! ColaErrorLento60segundos	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2LentoPetición-862366257	0	8	8	8	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción Consumo Inserción

Mostrando 1 a 9 de 9 Anterior | Siguiente

Paramos el servidor de aplicaciones.

Volvemos a arrancar el servidor de aplicaciones.

Vemos que los mensajes que no han sido tratados siguen en la cola, en este caso 2 mensajes:

Valores para wlsbJMSServer											
Configuración Registro Destinos Supervisión Control Notas											
Supervisión Destinos Activos Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas											
Esta página permite ver destinos activos dirigidos a este servidor JMS.											
Personalizar esta Tabla											
Destinos (Filtrado: Existen Más Columnas)											
Producción Consumo Inserción Mostrando 1 a 9 de 9 Anterior Siguiente											
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!ColaErrorLento60segundos	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2LentoPetition-862366257	0	2	2	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
Producción Consumo Inserción Mostrando 1 a 9 de 9 Anterior Siguiente											

Al arrancar el servidor se consumen los mensajes pendientes.

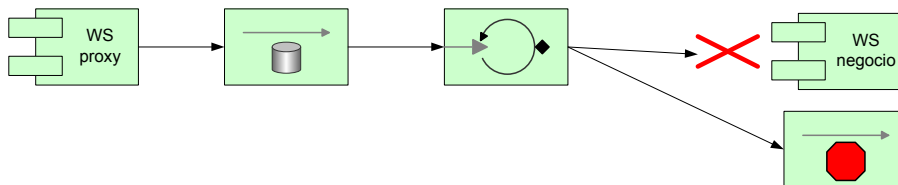
Esperamos 2 minutos y vemos que se han consumido los mensajes que quedaban.

<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!ColaErrorLento60segundos	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2LentoPetition-862366257	0	0	2	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
Producción Consumo Inserción Mostrando 1 a 9 de 9 Anterior Siguiente											

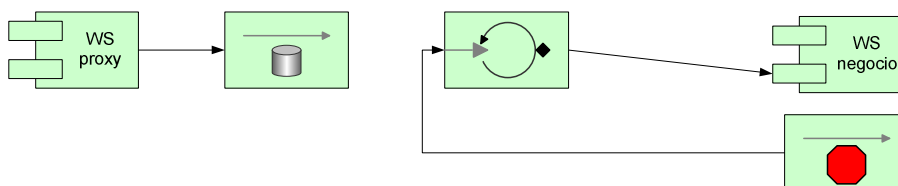
Caso de Uso 8: Canal de mensajes fallidos (dead letter cannel)

Alcance:

El caso de uso es una ampliación del canal de entrega garantizada. En primer lugar, cuando el consumidor de mensajes considere que un mensaje no se puede entregar al servicio de negocio (porque falla la conexión con el servicio o porque el propio servicio devuelve una excepción) el mensaje deberá guardarse en un canal de mensajes fallidos.



En segundo lugar, una vez que soporte y operación hayan resuelto el problema que hace que falle la entrega, se podrán reprocesar los mensajes del canal de mensajes fallidos.

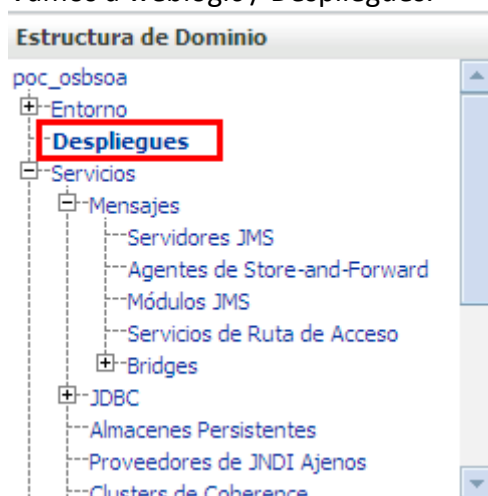


Se da por supuesto que el reproceso es una tarea de soporte y operación, es decir, incluirá tareas manuales de reconfiguración y ejecución del reproceso.

Se utiliza el mismo proyecto realizado en Canal de entrega garantizada.

Prueba

Vamos a weblogic / Despliegues.



Seleccionamos el wsdl que nos interesa, en este caso AAWSLento60segundos-WSLento60segundos-context-root.

Paramos el *Web Service*.

Instalar	Actualizar	Suprimir	Iniciar	Parar	Mostrando 11 a 20 de 167 Anterior Siguiente		
				Al terminar el trabajo Forzar Parada Ahora	Estado	Tipo	Orden de Despliegue
<input type="checkbox"/>	Nombre			Parar, pero continuar sirviendo solicitudes de administración	OK	Aplicación Web	100
<input type="checkbox"/>	AAWSBasadoContenido-WSServicio						
<input type="checkbox"/>	AAWSBasadoContenido-WSServicio33-context-root	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSDescripcion	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSDescripcionCompleta	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSLento-WSLento-context-root	Activo	OK			Aplicación Web	100
<input checked="" type="checkbox"/>	AAWSLento60segundos-WSLento60segundos-context-root	Activo	OK			Aplicación Web	100
Servicios Web							
	WSLento60segundosService					Servicio Web	
<input type="checkbox"/>	AAWSLento60segundos-WSLento-context-root	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSOrdenDePago	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSRegistro	Activo	OK			Aplicación Web	100
<input type="checkbox"/>	AAWSRegistro2	Activo	OK			Aplicación Web	100

De esta manera, al no estar disponible el servicio todos los mensajes que se envíen irán a la cola de error.

Hacemos varias llamadas desde el proyecto Canal de entrega garantizada, en este ejemplo se han realizado 3.

Recursos			
Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 2 de 2			
<input type="checkbox"/>	Nombre	Tipo de Recurso	Acciones
<input type="checkbox"/>	PS1	Servicio de Proxy	
<input type="checkbox"/>	PS2	Servicio de Proxy	
Elementos 1 - 2 de 2			
Suprimir			

Todos los mensajes van a la cola de error.

Destinos (Filtrado: Existen Más Columnas)											
Producción	Consumo	Inserción	Mostrando 1 a 9 de 9 Anterior Siguiente								
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources!ColaErrorLento60segundos	3	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!PS2LentoPetición352247522	0	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
Producción	Consumo	Inserción	Mostrando 1 a 9 de 9 Anterior Siguiente								

Volvemos a iniciar el wsdl que hemos parado.

Despliegues

Mostrando 11 a 20 de 168 Anterior | Siguiente

Nombre	Estado	Estado	Tipo	Orden de Despliegue
<input type="checkbox"/> AAWSBasadoContenido-WSServicio2-context-root	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSBasadoContenido-WSServicio33-context-root	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSDescripcion	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSDescripcionCompleta	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSLento-WSLento-context-root	Activo	✓ OK	Aplicación Web	100
<input checked="" type="checkbox"/> AAWSLento60segundos-WSLento60segundos-context-root	Preparado	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSLentoooo-WSLento-context-root	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSOrdenDePago	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSRegistro	Activo	✓ OK	Aplicación Web	100
<input type="checkbox"/> AAWSRegistro2	Activo	✓ OK	Aplicación Web	100

Mostrando 11 a 20 de 168 Anterior | Siguiente

Accedemos a Servicios / Mensajes / ServidoresJMS.



Pinchamos en servidor JMS wlsbjMSServer.

Servidores JMS (Filtrado: Existen Más Columnas)

Mostrando 1 a 6 de 6 Anterior | Siguiente

Nombre	Almacén Persistente	Destino	Servidor Actual	Estado
<input type="checkbox"/> BPMJMServer	BPMJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/> JMSserver	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/> SOAJMSserver	SOAJMSFileStore	soa_server1	soa_server1	
<input type="checkbox"/> UMSJMSserver	UMSJMSFileStore	soa_server1	soa_server1	
<input checked="" type="checkbox"/> wlsbjMSServer	FileStore	osb_server1	osb_server1	✓ OK
<input type="checkbox"/> WseeJmsServer	WseeFileStore	osb_server1	osb_server1	✓ OK

Mostrando 1 a 6 de 6 Anterior | Siguiente

Supervisión / Destinos Activos. Seleccionamos la cola de peticiones, en este caso PS2LentoPetición y paramos el consumo.

Valores para wlsbjMSServer

Configuración Registro Destinos **Supervisión** Control Notas

Supervisión **Destinos Activos** Transacciones Activas Conexiones Activas Pools de Sesiones Activas Conexiones de Pool Activas

Esta página permite ver destinos activos dirigidos a este servidor JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Producción Consumo Inserción

Mostrando 1 a 9 de 9 Anterior Siguiente

	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources! ColaErrorLento60segundos	3	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false
<input checked="" type="checkbox"/>	jmsResources! PS2LentoPetición352247522	0	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources/wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false

Producción Consumo Inserción

Mostrando 1 a 9 de 9 Anterior Siguiente

Accedemos a Servicios / Mensajes / Módulos JMS.

Estructura de Dominio

- poc_osboa
 - Entorno
 - Despliegues
 - Servicios
 - Mensajes
 - Servidores JMS
 - Agentes de Store-and-Forward
 - Módulos JMS**
 - Servicios de Ruta de Acceso
 - Bridges
 - JDBC
 - Almacenes Persistentes
 - Proveedores de JNDI Ajenos
 - Clusters de Coherencia

Pinchamos en Módulo JMS jmsResources.

Módulos JMS

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior Siguiente

Nombre	Tipo
BPMJMSModule	Sistema
jmsResources	Sistema
SOAJMSModule	Sistema
SystemModule	Sistema
UMSJMSSystemResource	Sistema
WseeJmsModule	Sistema

Nuevo Suprimir

Mostrando 1 a 6 de 6 Anterior Siguiente

“ColaErrorLento60segundos”.

Resumen de Recursos

Nuevo Suprimir

Mostrando 1 a 10 de 12 Anterior | Siguiente

<input type="checkbox"/>	Nombre	Tipo	Nombre de JNDI	Despliegue Secundario	Destinos
<input type="checkbox"/>	ColaErrorLento60segundos	Cola	ColaErrorLento60segundos	ColaErrorLento60segundos	wlsbJMSServer
<input type="checkbox"/>	PS2LentoPetición352247522	Cola	PS2LentoPetición	PS2LentoPetición352247522	wlsbJMSServer
<input type="checkbox"/>	TemporaryTmplt	Plantilla	N/A	N/A	N/A
<input type="checkbox"/>	weblogic.wlsb.jms.transport.task.QueueConnectionFactory	Fábrica de Conexiones	weblogic.wlsb.jms.transport.task.QueueConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/>	wli.reporting.jmsprovider.ConnectionFactory	Fábrica de Conexiones	wli.reporting.jmsprovider.ConnectionFactory	Direccionamiento por Defecto	osb_server1
<input type="checkbox"/>	wli.reporting.jmsprovider.queue	Cola	wli.reporting.jmsprovider.queue	wlsbJMSServer	wlsbJMSServer
<input type="checkbox"/>	wli.reporting.jmsprovider_error.queue	Cola	wli.reporting.jmsprovider_error.queue	wlsbJMSServer	wlsbJMSServer
<input type="checkbox"/>	wli.reporting.purge.queue	Cola	wli.reporting.purge.queue	wlsbJMSServer	wlsbJMSServer
<input type="checkbox"/>	wlsb.internal.transport.task.queue.email	Cola	wlsb.internal.transport.task.queue.email	wlsbJMSServer	wlsbJMSServer
<input type="checkbox"/>	wlsb.internal.transport.task.queue.file	Cola	wlsb.internal.transport.task.queue.file	wlsbJMSServer	wlsbJMSServer

Nuevo Suprimir

Mostrando 1 a 10 de 12 Anterior | Siguiente

Supervisión / seleccionamos la cola / Mostrar mensajes.

Valores para ColaErrorLento60segundos

Configuración **Supervisión** Control Seguridad Despliegue Secundario Notas

Un destino de JMS identifica una cola (punto a punto) o un tema (publicación/suscripción) dirigidos a un servidor JMS.
En esta página se resumen los destinos de JMS activos que se han creado para este módulo JMS.

Personalizar esta Tabla

Destinos (Filtrado: Existen Más Columnas)

Mostrar Mensajes

Mostrando 1 a 1 de 1 Anterior | Siguiente

<input checked="" type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Mensajes Totales	Consumidores Actuales	Recuento Alto de Consumidores	Consumidores Totales	Recuento Alto de Mensajes
<input checked="" type="checkbox"/>	jmsResources! ColaErrorLento60segundos	3	0	3	0	0	0	3

Mostrar Mensajes

Mostrando 1 a 1 de 1 Anterior | Siguiente

Mover / Mover todo (o sólo los seleccionados)

Mensajes de JMS (Filtrado: Existen Más Columnas)

Nuevo Suprimir **Mover** Importar Exportar

Mostrando 1 a 3 de 3 Anterior | Siguiente

<input type="checkbox"/>	Identificador	Identificador de Correlación	Registro de Hora	Cadena de Estado	Modo de Entrega de JMS	Tamaño del Mensaje
<input type="checkbox"/>	ID: <589818.1295873213391.0>		Mon Jan 24 13:46:53 CET 2011	visible	Persistente	304
<input type="checkbox"/>	ID: <589818.1295873219629.0>		Mon Jan 24 13:46:59 CET 2011	visible	Persistente	304
<input type="checkbox"/>	ID: <589818.1295873221986.0>		Mon Jan 24 13:47:01 CET 2011	visible	Persistente	304

Nuevo Suprimir Mover Importar Exportar

Mostrando 1 a 3 de 3 Anterior | Siguiente

Elegimos el servidor JMS donde se encuentran las colas, wlsbJMSServer. Siguiente.

Mover Mensajes de JMS

Atrás Siguiente Terminar Cancelar

Mover Mensajes de JMS

Seleccione el servidor JMS que aloja el destino final en el que se van a reubicar los mensajes.

Servidor JMS: wlsbJMSServer

Atrás Siguiente Terminar Cancelar

Elegimos que los mueva a la cola de peticiones, en este caso PS2LentoPetición. Terminar.

Mover Mensajes de JMS

Atrás | Siguiente | Terminar | Cancelar

Mover Mensajes de JMS
 Seleccione el destino final en el que se reubicarán los mensajes.

Destino: jmsResources!PS2LentoPetición352247522

Atrás | Siguiente | Terminar | Cancelar

Si volvemos a Servicios / Mensajes / Servidores JMS / wlsbJMSServer / Supervisión /Destinos Activos, vemos que los tres mensajes de la cola de errores se han pasado a la cola de peticiones.

Destinos (Filtrado: Existen Más Columnas)

Producción			Consumo			Inserción			Mostrando 1 a 9 de 9 Anterior Siguiente			
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa	
<input type="checkbox"/>	jmsResources!ColaErrorLento60segundos	0	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!PS2LentoPetición352247522	3	0	3	6	0	Queue	paused_consumption, advertised_in_cluster_jndi	false	false	true	
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
Producción			Consumo			Inserción			Mostrando 1 a 9 de 9 Anterior Siguiente			

Reiniciamos el consumo de la cola PS2LentoPetición.

Destinos (Filtrado: Existen Más Columnas)

Producción			Consumo			Inserción			Mostrando 1 a 9 de 9 Anterior Siguiente			
<input type="checkbox"/>	Nombre	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa	
<input type="checkbox"/>	jmsResources!ColaErrorLento60segundos	0	0	3	3	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input checked="" type="checkbox"/>	jmsResources!PS2LentoPetición352247522	3	0	3	6	0	Queue	paused_consumption, advertised_in_cluster_jndi	false	false	true	
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wli.reporting.jmsprovider_error.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wli.reporting.purge.queue	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
<input type="checkbox"/>	jmsResources!wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_jndi	false	false	false	
Producción			Consumo			Inserción			Mostrando 1 a 9 de 9 Anterior Siguiente			

Como el Web Service ya está disponible, los tres mensajes son tratados.

CASOS DE USO DEL GAD

Caso de Uso 1: “Invocación de WS con tipos de datos propietarios de .net desde Java

Alcance:

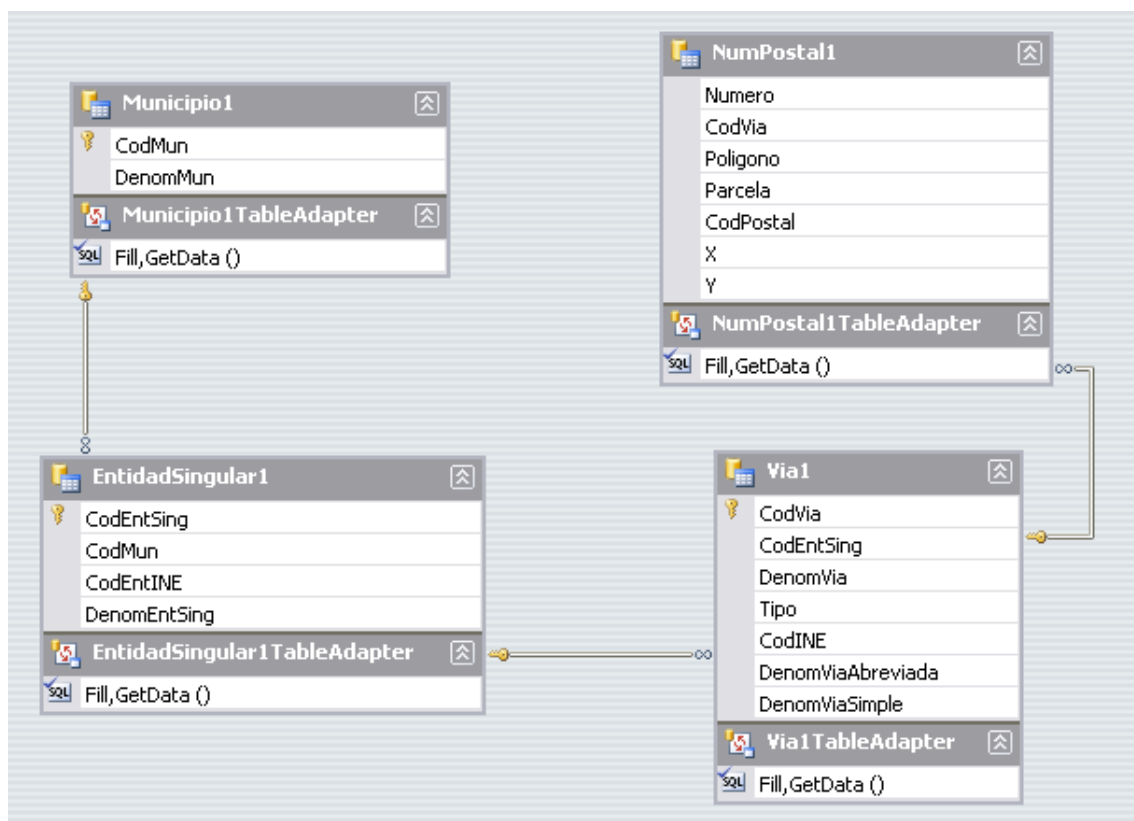
Existen ciertos WS ya desarrollados en GN en tecnología .Net que devuelven tipos de datos no consumibles directamente desde Java y que implican un redesarrollo de dichos WS, no siendo siempre factible este redesarrollo.

El caso de prueba debe seguir y cubrir lo siguiente:

- Crear un WS con las mismas especificaciones que el WS real (GDP) aunque siempre devuelva el mismo dato
- Publicación del mismo en el Bus e intento de consumirlo y probar a cambiar el tipo si es factible (utilizar todos los recursos de soporte disponibles)
- En caso de no ser posible proponer una alternativa al mismo WS con datos compatibles que devuelva el mismo dato, pero de un modo alternativo y diferente

En última instancia, y sólo si está disponible el WS GDP sin datos “no estándar” (en desarrollo actualmente), se intentará publicar esa nueva versión “estándar” en el Bus a modo de validación funcional de que el Bus permitirá su publicación.

Estructura WS .Net (DataSet)



RESUMEN PRUEBAS REALIZADAS

	SATISFACTORIO	NO SATISFACTORIO
Llamada directa desde Java a un WS .Net con DataSet.		✓
Llamada directa desde .Net a un WS .Net con DataSet.	✓	
Llamada desde Java al OSB dónde está publicado el WS .Net con DataSet.		✓
Llamada desde .Net al OSB dónde está publicado el WS .Net con DataSet.	✓	
Llamada desde Java al OSB dónde está publicado el WS .Net con DataSet y se han realizado las transformaciones necesarias.	✓	
Llamada desde .Net al OSB dónde está publicado el WS .Net con DataSet y se han realizado las transformaciones necesarias.	✓	

PASO 1-LLAMADA DIRECTA

Creamos un nuevo proyecto llamado “InvocacionDataSet”. Añadimos tres carpetas: *Business Service*, *Proxy Service* y *Recursos*.

En la carpeta *Recursos* añadimos el wsdl de .Net:

- Service1.wsdl

<http://a.b.c.d/prueba/Service1.asmx?wsdl>

En la carpeta *Business Service* creamos un nuevo servicio de negocio.



Indicamos un nombre “Bs_DSGDP” y lo creamos a partir del wsdl de .Net.

Crear un Servicio de Negocio (InvocacionDataSet/Business Service/)

Configuración General

Nombre del Servicio* Bs_DSGDP

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☒ Servicio Web WSDL (puerto)

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☐ Servicio de Negocio

☐ Servicio de Proxy

“Último” y “Guardar”.

Creamos un *Proxy Service* llamado “PS_DSGDP” a partir del servicio de negocio que acabamos de crear:

InvocacionDataSet/Proxy Service

Referencias

Referencias de otro

Carpetas

Introducir Nombre

Nombre

Suprimir

Recursos

Crear Recurso: Seleccionar Tipo de Recurso

Crear un Servicio de Proxy (InvocacionDataSet/Proxy Service/)

Configuración General

Nombre del Servicio* PS_DSGDP

Descripción

Tipo de Servicio*

Crear un Servicio Nuevo

☐ Servicio Web WSDL

☐ Servicio de Transporte según Configuración

☐ Servicio de Mensajes

☐ Cualquier Servicio SOAP SOAP 1.1

☐ Cualquier Servicio XML

Crear a partir de Servicio Existente

☒ Servicio de Negocio InvocacionDataSet/Business Service/BS_DSGDP

☐ Servicio de Proxy

Siguiente >> Último >> Cancelar

“Guardar todo” y “Activar sesión”

Prueba en la consola del OSB

Ejecutamos el PS_DSGDP de la consola del OSB.

Recursos			
Crear Recurso: Seleccionar Tipo de Recurso			
Elementos 1 - 1 de 1			
Nombre	Tipo de Recurso	Acciones	Opciones
PS_DSGDP	Servicio de Proxy		aTe
Elementos 1 - 1 de 1			
Suprimir			

Obtenemos la siguiente respuesta:

Documento de Respuesta

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soap:Body xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <mostrarDatosResponse xmlns="http://tempuri.org/">
      <mostrarDatosResult>
        <xs:schema id="DSGDP" targetNamespace="http://www.tracasa.es/Schemas/GDP/DSGDP.xsd" attributeFormDefault="qualified" elementFormDefault="qualified"
          xmlns:mstns="http://www.tracasa.es/Schemas/GDP/DSGDP.xsd" xmlns="http://www.tracasa.es/Schemas/GDP/DSGDP.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
          <xs:element name="DSGDP" msdata:IsDataSet="true" msdata:Locale="en-US">
            <xs:complexType>
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="Municipio">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="CodMun" type="xs:short"/>
                      <xs:element name="DenomMun" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="EntidadSingular">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="CodEntSing" type="xs:short"/>
                      <xs:element name="CodMun" type="xs:short"/>
                      <xs:element name="CodEntINE" type="xs:short"/>
                      <xs:element name="DenomEntSing" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="Via">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="CodVia" type="xs:int"/>
                      <xs:element name="CodEntSing" type="xs:short"/>
                      <xs:element name="DenomVia" type="xs:string" minOccurs="0"/>
                      <xs:element name="Tipo" type="xs:string" minOccurs="0"/>
                      <xs:element name="CodINE" type="xs:int" default="0" minOccurs="0"/>
                      <xs:element name="DenomViaAbreviada" type="xs:string" minOccurs="0"/>
                      <xs:element name="DenomViaSimple" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="NumPostal">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Numero" type="xs:string" minOccurs="0"/>
                      <xs:element name="CodVia" type="xs:int"/>
                      <xs:element name="Poligono" type="xs:short" minOccurs="0"/>
                      <xs:element name="Parcela" type="xs:short" minOccurs="0"/>
                      <xs:element name="CodPostal" type="xs:int" minOccurs="0"/>
                      <xs:element name="X" type="xs:decimal" default="0.00" minOccurs="0"/>
                      <xs:element name="Y" type="xs:decimal" default="0.00" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="NumPostal1">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Numero" type="xs:string" minOccurs="0"/>
                      <xs:element name="CodVia" type="xs:int" minOccurs="0"/>
                      <xs:element name="Poligono" type="xs:int" minOccurs="0"/>
                      <xs:element name="Parcela" type="xs:int" minOccurs="0"/>
                      <xs:element name="CodPostal" type="xs:int" minOccurs="0"/>
                      <xs:element name="X" type="xs:int" minOccurs="0"/>
                      <xs:element name="Y" type="xs:int" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:unique name="Constraint1" msdata:PrimaryKey="true">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodMun"/>
          </xs:unique>
          <xs:unique name="Constraint2">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodEntSing"/>
          </xs:unique>
          <xs:unique name="DSGDPKey1" msdata:PrimaryKey="true">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodVia"/>
          </xs:unique>
          <xs:keyref name="ViaNumPostal" refer="DSGDPKey1">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodVia"/>
          </xs:keyref>
          <xs:keyref name="EntidadSingularVia" refer="Constraint2">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodEntSing"/>
          </xs:keyref>
          <xs:keyref name="nomen_municipionomen_ent_singular" refer="Constraint1">
            <xs:selector xpath="."/>
            <xs:field xpath="mstns:CodMun"/>
          </xs:keyref>
        </xs:element>
      </mostrarDatosResult>
    </mostrarDatosResponse>
  </soap:Body>
</soap:Envelope>

```

```

</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <DSGDP xmlns="http://www.tracasa.es/Schemas/GDP/DSGDP.xsd">
    <NumPostal1 diffgr:id="NumPostal11" msdata:rowOrder="0">
      <Numero>54</Numero>
      <CodVia>2</CodVia>
      <Poligono>3</Poligono>
      <Parcela>44</Parcela>
      <CodPostal>31011</CodPostal>
      <X>254</X>
      <Y>322</Y>
    </NumPostal1>
    <NumPostal1 diffgr:id="NumPostal12" msdata:rowOrder="1">
      <Numero>28</Numero>
      <CodVia>3</CodVia>
      <Poligono>5</Poligono>
      <Parcela>45</Parcela>
      <CodPostal>31011</CodPostal>
      <X>248</X>
      <Y>625</Y>
    </NumPostal1>
  </DSGDP>
</diffgr:diffgram>
</mostrarDatosResult>
</mostrarDatosResponse>
</soap:Body>
</soapenv:Envelope>

```

Metadatos de Respuesta

```

<con:metadata xmlns:con="http://www.bea.com/wli/sb/test/config">
  <tran:headers xsi:type="http:HttpResponseHeaders" xmlns:http="http://www.bea.com/wli/sb/transports/http" xmlns:tran="http://www.bea.com/wli/sb/transports"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <http:Content-Type>text/xml; charset=utf-8</http:Content-Type>
  </tran:headers>
  <tran:response-code xmlns:tran="http://www.bea.com/wli/sb/transports">0</tran:response-code>
  <tran:encoding xmlns:tran="http://www.bea.com/wli/sb/transports">utf-8</tran:encoding>
</con:metadata>

```

Rastreo de Llamada

- (recibiendo petición)
- RouteTo_BS_DSGDP

Parece que el OSB ha realizado las transformaciones necesarias pero se ha creado un cliente en Java (ClienteDataSet.zip) consumiendo este wsdl publicado por el OSB y nos encontramos en el mismo caso que si no utilizamos el bus como intermediario, es decir, recibe un xml pero no sabe como tratarlo.

PASO 2-TRANSFORMACIÓN EN EL OSB

Este proyecto es necesario crearlo en Eclipse ya que nos facilita la forma de crear las transformaciones XQuery.

Creamos un nuevo proyecto llamado “Prueba”. Añadimos tres carpetas: *Business Service*, *Proxy Service* y Recursos.

En la carpeta recursos añadimos dos wsdl:

- LlamadaMunicipioService.wsdl

<http://a.b.c.d:8081/Prueba1-Prueba-context-root/LlamadaMunicipioPort?WSDL>

- Service1.wsdl

<http://a.b.c.d/prueba/Service1.asmx?wsdl>

En este ejemplo el Proxy hace referencia al wsdl LlamadaMunicipioService.wsdl

Se trata de un *Web Service* en Java creado con los mismos objetos del DataSet.

Creamos un nuevo Proxy en la carpeta *Proxy Service*. Le damos el nombre PS_LlamadaMunicipios.proxy.

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type*

Create a New Service

☒ WSDL Web Service

SOAP 1.1 (port)

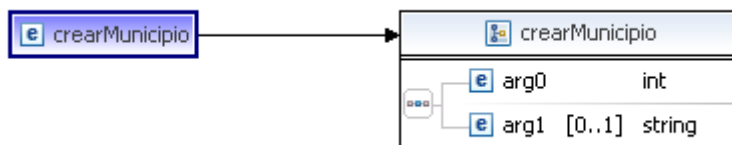
☐ Transport Typed Service

☐ Messaging Service

☐ Any SOAP Service

☐ Any XML Service

Los datos que se piden como entrada son:



Se ha creado un *Web Service* llamado service1.wsdl en .Net que sigue las mismas especificaciones que el WS real (GDP).

En la carpeta *Business Service* creamos un nuevo servicio de negocio llamado BS_DSGDP que hace referencia al *Web Service* creado en .Net.

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type *

Create a New Service

☒ WSDL Web Service

SOAP 1.1 (port)

☐ Transport Typed Service

☐ Messaging Service

☐ Any SOAP Service

☐ Any XML Service

Accedemos al flujo de mensajes del Proxy que hemos creado:

General Configuration

Configuration

Use this page to edit the general information for this service.

Description

Service Type *

Create a New Service

☒ WSDL Web Service

SOAP 1.1 (port)

☐ Transport Typed Service

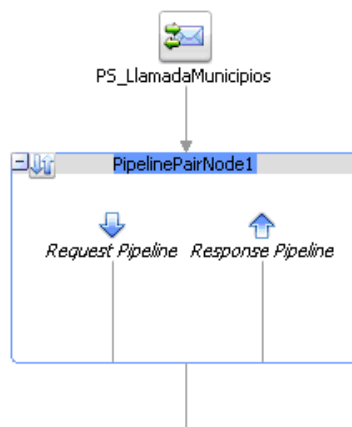
☐ Messaging Service

☐ Any SOAP Service

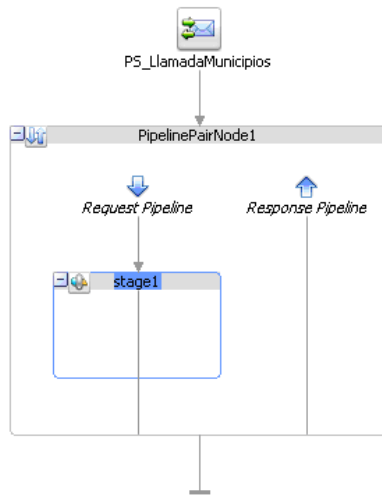
☐ Any XML Service

General | Transport | HTTP Transport | Operation | Message Handling | Policy | Security | **Message Flow**

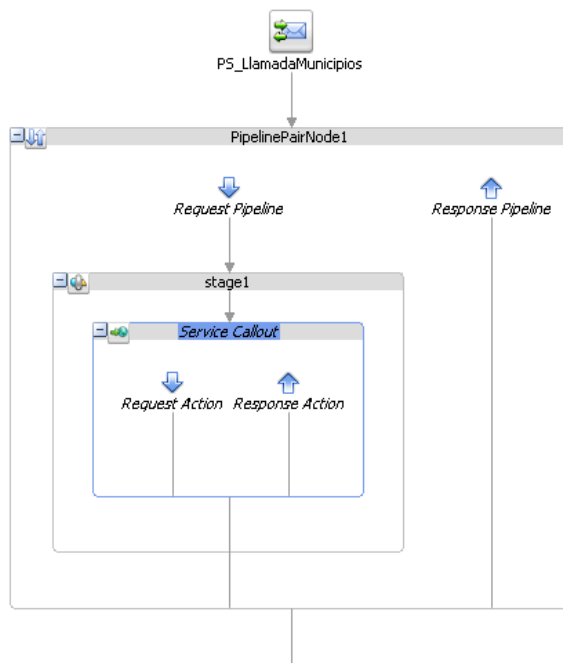
Añadimos un nuevo "Pipeline pair".



Dentro del "pipeline pair" añadimos una nueva etapa "stage".



Añadimos un nuevo “Service Callout” para invocar al servicio que contiene el *Web Service* de .Net.



Indicamos que la llamada la haga al servicio de negocio BS_DSGDP y elegimos la única operación existente mostrarDatos. Este método no necesita parámetros de entrada pero es obligatorio indicar una variable de entrada, ponemos por ejemplo 'body' que ya se encuentra inicializada. Como variable de salida (donde se va a guardar el resultado de la invocación) lo ponemos en la variable 'Salida', por ejemplo.

Service Callout

Service: * Prueba/Business Service/BS_DSGDP [Browse]

Invoking: * mostrarDatos [v]

☐ Configure Soap Body ☒ Configure Payload Document

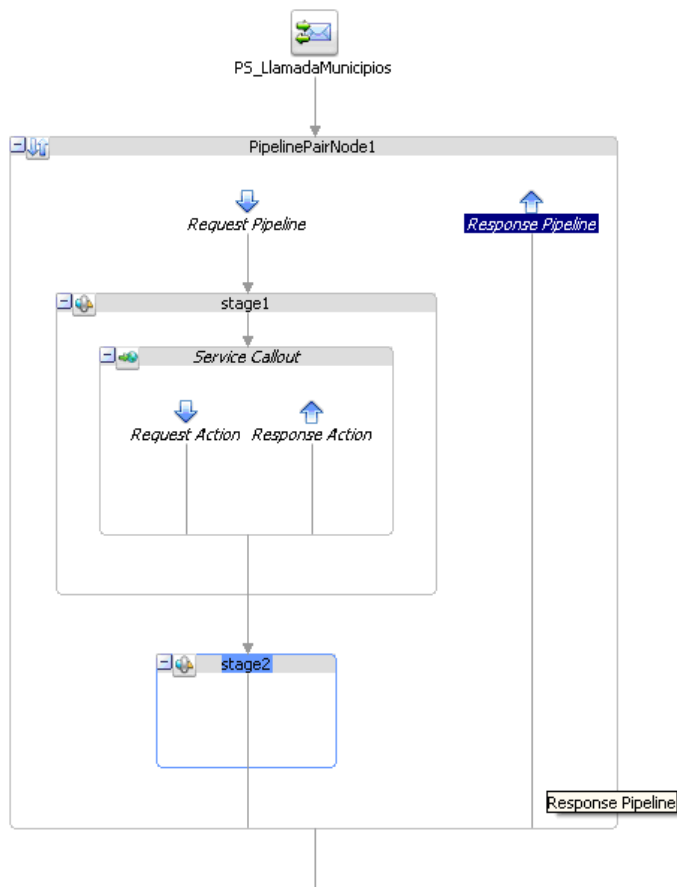
Request Variable: body

Response Variable: Salida

Request Header: []

Response Header: []

Una vez que hemos realizado la invocación al servicio de .Net, añadimos una nueva etapa "Stage".



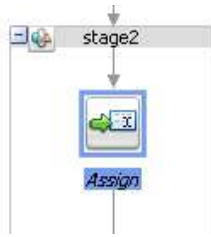
Si realizamos una llamada directa al servicio de negocio de .Net, nos devuelve lo mismo que en el Paso1. Lo que nos devuelve es lo siguiente:

De todo este xml, lo único que nos interesa es la etiqueta "NumPostal1".

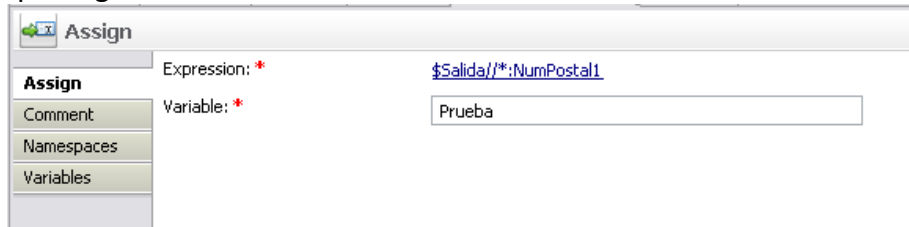
```

<NumPostal1 diffgr:rid="NumPostal11" msdata:rowOrder="0">
  <Numero>54</Numero>
  <CodVia>2</CodVia>
  <Poligono>3</Poligono>
  <Parcela>44</Parcela>
  <CodPostal>31011</CodPostal>
  <X>254</X>
  <Y>322</Y>
</NumPostal1>
  
```

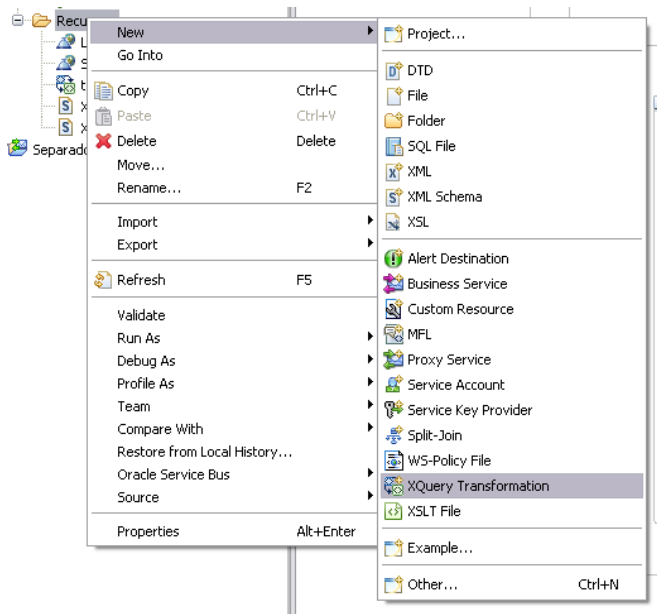
Por ello, añadimos un nuevo "Assign" dentro de la etapa que acabamos de crear.



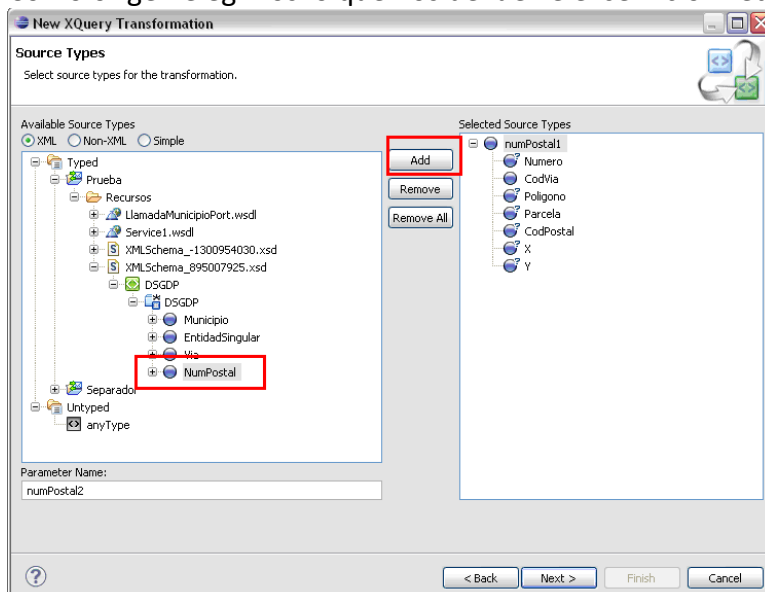
Le indicamos que saque de la variable “\$Salida” la parte del xml que nos hace falta y que la guarde en la variable “Prueba”.



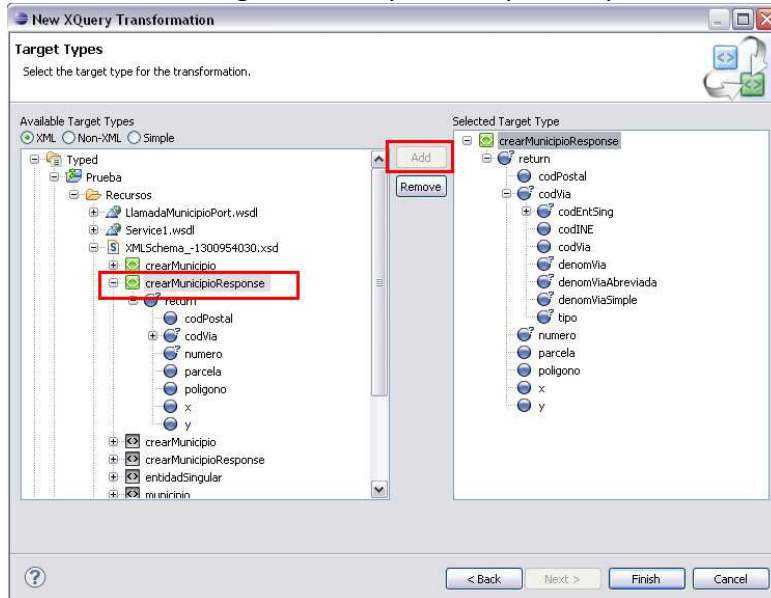
En la carpeta Recursos creamos una nueva transformación XQuery llamada ‘transform’.



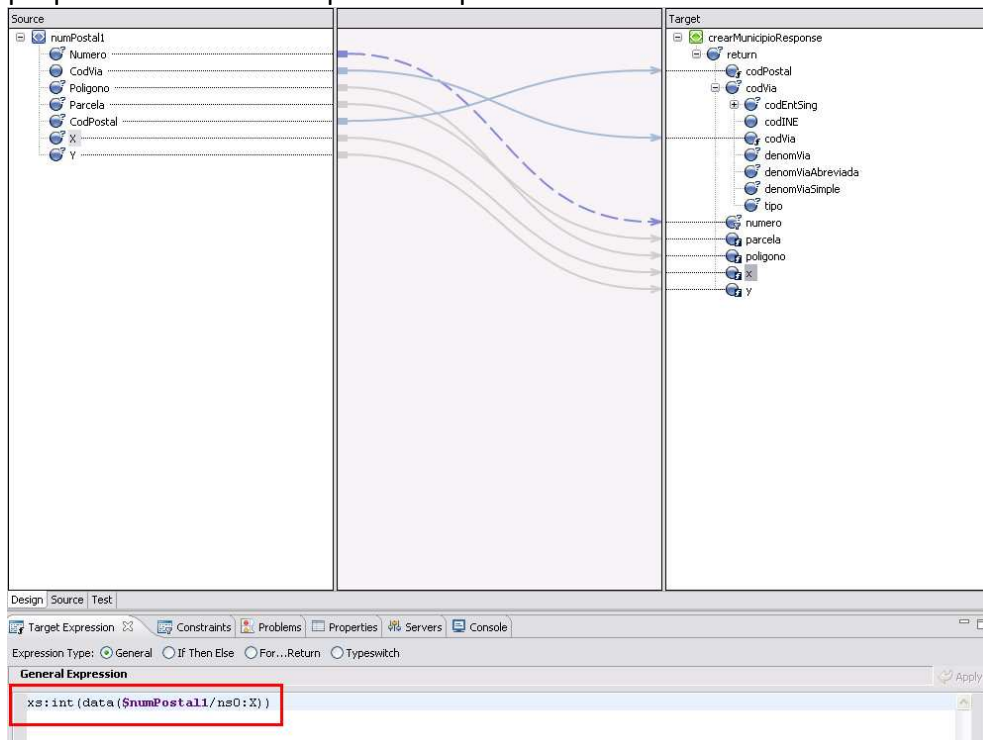
Como origen elegimos lo que nos devuelve el servicio .Net, en este caso NumPostal.



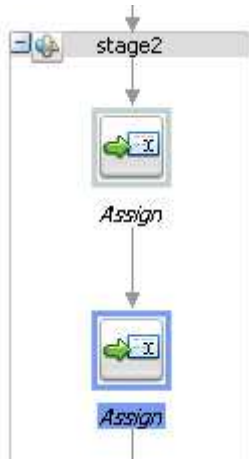
Como destino elegimos el esquema esperado por el wsdl de Java.



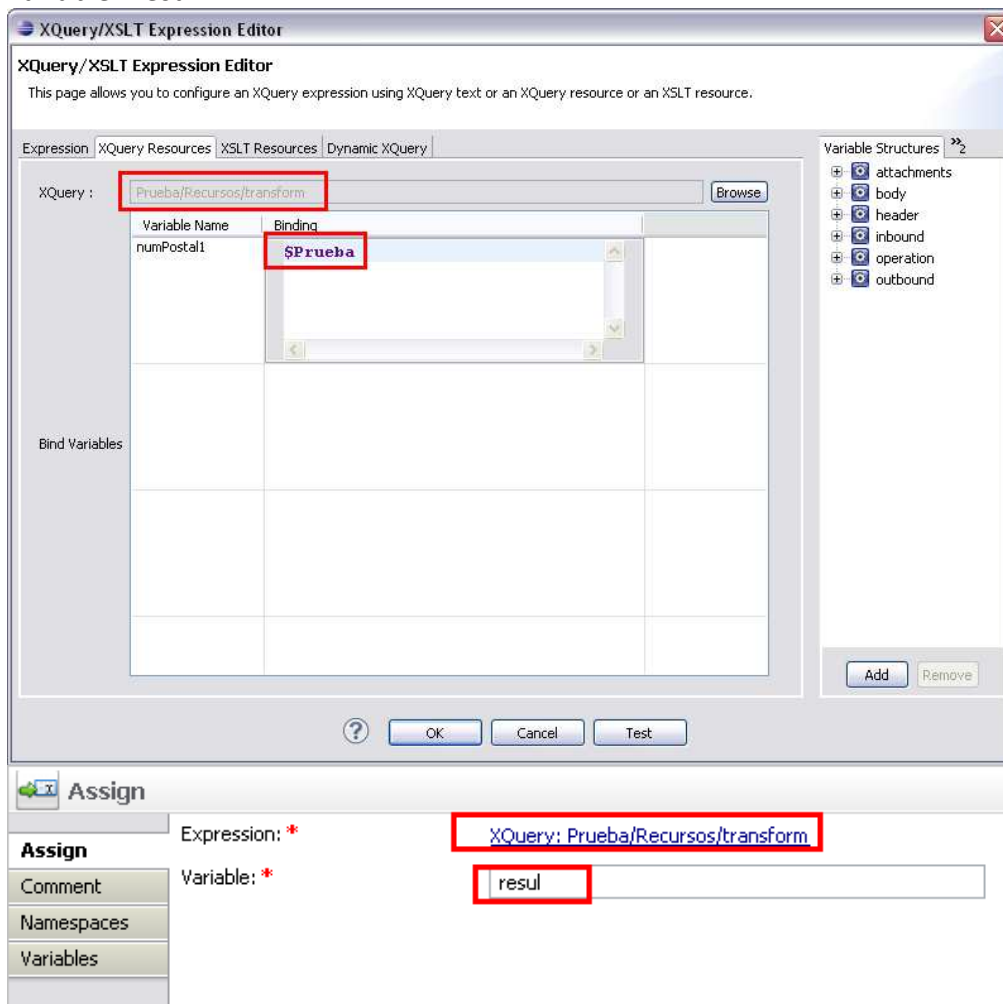
Unimos cada uno de los campos de origen y destino. Si se trata del mismo tipo, los unimos sin problemas. Hay algunos casos como por ejemplo el campo X es de tipo decimal y el campo x es de tipo int. Para ello pinchamos en el campo x y en propiedades indicamos que el campo X lo convierta en int:



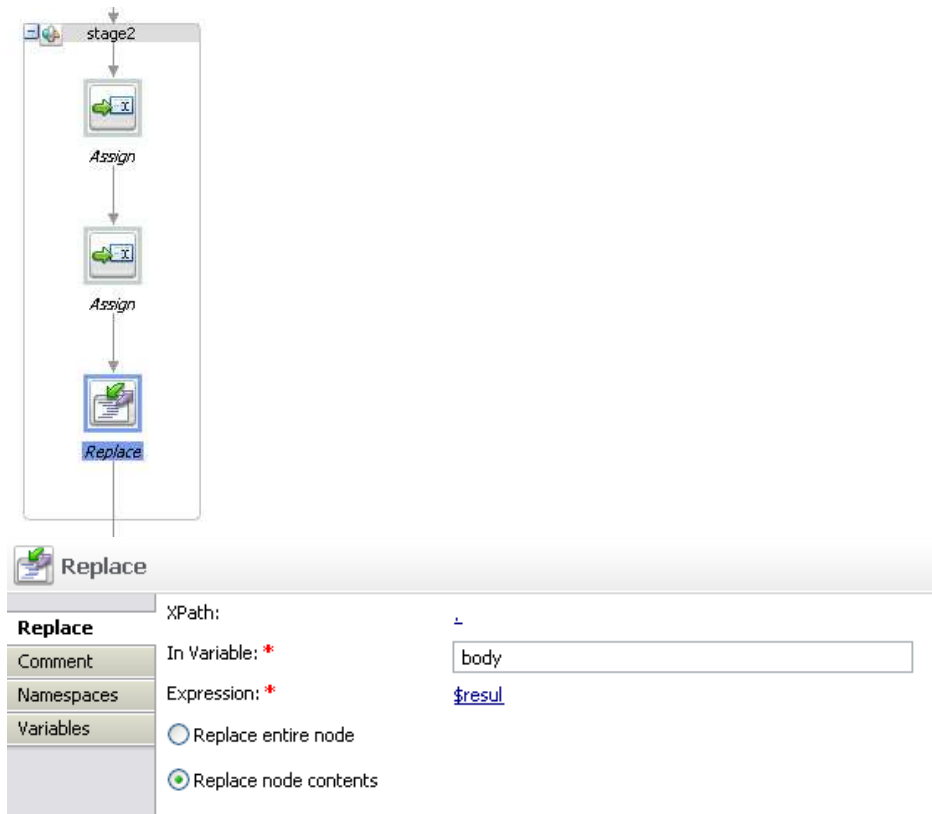
Añadimos un nuevo "Assign":



En el campo “Expression” indicamos la transformación XQuery y como origen los valores que se encuentran en la variable “\$Prueba”. Guardamos el resultado en la variable “result”.

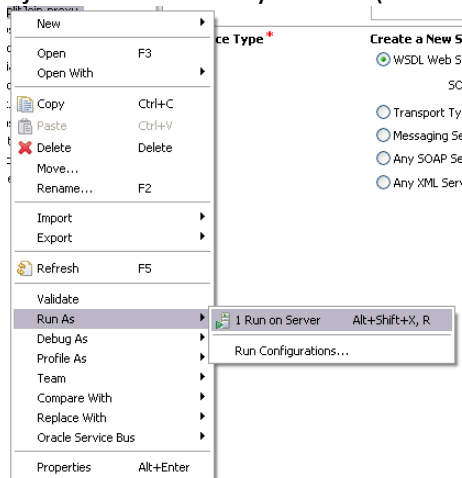


Añadimos un “Replace” para sustituir la variable \$body por la variable \$result.



Prueba en la consola del OSB

Ejecutamos el Proxy creado (botón derecho sobre el Proxy Run As/Run on Server):



Introducimos un int y un string cualquiera ya que no va a influir en la petición.

Prueba de Servicios de Proxy - PS_LlamadaMunicipios Ayuda

Operaciones Disponibles: crearMunicipio

Ejecutar Ejecutar-Guardar Restablecer Cerrar

Configuración de Prueba

Llamada Directa: ☒
Incluir Rastreo: ☒

Documento de Petición

Pantalla XML

Cabecera SOAP:

```
<soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
</soap:Header>
```

Carga Útil: Examinar...

```
<pru:crearMunicipio xmlns:pru="http://prueba/">
  <arg0>1</arg0>
  <!--Optional:-->
  <arg1>Pamplona</arg1>
</pru:crearMunicipio>
```

Obtenemos el resultado esperado, es decir, la respuesta esperada por el WS Java pero con los datos devueltos por el WS .Net.

Prueba de Servicios de Proxy - PS_LlamadaMunicipios Ayuda

Atrás Cerrar

Documento de Petición

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <pru:crearMunicipio xmlns:pru="http://prueba/">
      <arg0>1</arg0>
      <!--Optional:-->
      <arg1>Pamplona</arg1>
    </pru:crearMunicipio>
  </soapenv:Body>
</soapenv:Envelope>
```

Documento de Respuesta

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  </soap:Header>
  <soapenv:Body>
    <pru:crearMunicipioResponse xmlns:pru="http://prueba/">
      <return>
        <codPostal>31011</codPostal>
        <codVia>
          <codVia>2</codVia>
        </codVia>
        <numero>54</numero>
        <parcela>44</parcela>
        <poligono>3</poligono>
        <x>254</x>
        <y>322</y>
      </return>
    </pru:crearMunicipioResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Metadatos de Respuesta

Realizamos la llamada desde un cliente Java y funciona de manera correcta (Prueba llamada con transformación/ClienteJava.zip).

Anexo III: Entregables de diseño

Informes con *Oracle Service Bus*

MONITORIZACIÓN Y PRESENTACIÓN DE INFORMES

La monitorización es la capacidad de recoger información en tiempo de ejecución, mientras que la presentación de informes se centra en la entrega de datos de mensajes y alertas mediante informes.

MONITORIZACIÓN

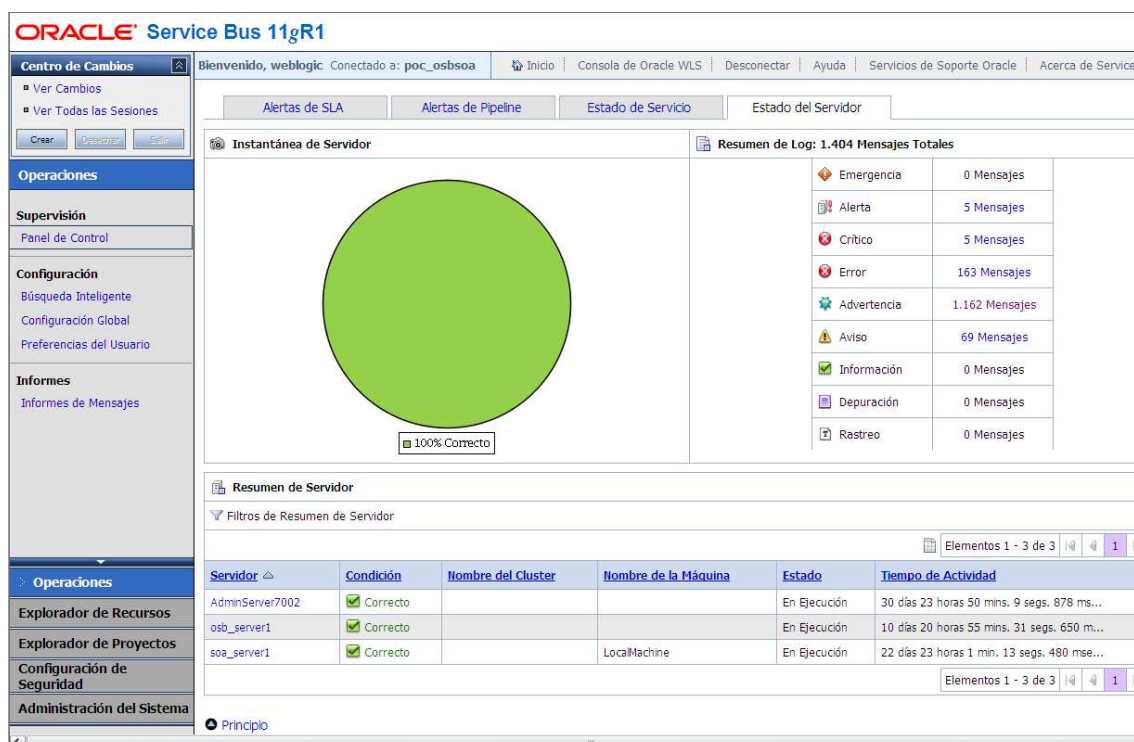
La monitorización se centra en la información operativa: el estado general del bus de servicios, cómo se han generado muchas de las alertas y de qué tipo son, la situación de los SLA y la recopilación de las estadísticas relacionadas.

El **panel de control** nos permite tener una vista predeterminada del bus de servicios. Nos hace un resumen de las alertas SLA, alertas de error (manejador de errores), estado de servicio y estado del servidor.

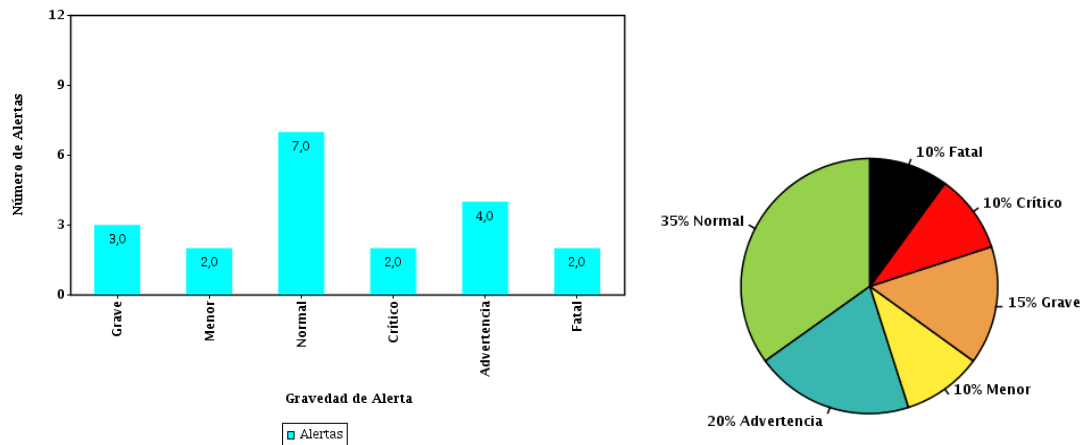
Mediante un gráfico circular se muestra el estado del servidor. Dicho gráfico es de color verde si el servidor (único en este caso) se está ejecutando bien.

En la parte derecha del panel de control se encuentra el resumen de Log, que muestra el número y la gravedad de todas las alertas que han entrado en el registro de alertas.

En la parte inferior del panel de control se muestra una descripción más detallada de la situación específica del servidor, incluyendo los nombres del clúster y la máquina de cada servidor.



En las pestañas Alertas de SLA y Alertas de error se muestra las alertas existentes así como estadísticas del número y tipo de alertas:



La pestaña estado de servicio presenta un resumen del tiempo de servicio, número de alertas SLA, número de alertas de error, tiempo de respuesta, número de mensajes entrantes, etc.

DESTINOS DE ALERTAS

Oracle Service Bus permite configurar el envío de las alertas a traps SNMP, un informe, una dirección de correo electrónico, una cola JMS, a la consola o una combinación de cualquiera de estos destinos, con el fin de notificar diferentes incidencias producidas durante la ejecución y mediación de los servicios desplegados en la plataforma. De este modo, es posible integrar la información de monitorización de OSB en sistemas de terceros.

Edit Alert Destination - Prueba/Proxy Services/Alerta Crítica	
Resource Name*	Alerta Crítica
Resource Description	
SNMP Trap	<input checked="" type="radio"/> Yes <input type="radio"/> No
Reporting	<input checked="" type="radio"/> Yes <input type="radio"/> No
Alert Logging	<input checked="" type="radio"/> Yes <input type="radio"/> No
e-mail Recipients	<div>Recipients Options</div> <div>No email recipients to display.</div> <div>Add</div>
JMS Destinations	<div>Destinations Options</div> <div>No JMS destinations to display.</div> <div>Add</div>
<div>Save Cancel</div>	

SNMP

Oracle Service Bus permite el envío de traps SNMP con el fin de notificar diferentes incidencias producidas durante la ejecución y mediación de los servicios desplegados

en la plataforma. De este modo, es posible integrar la información de monitorización de OSB en sistemas de terceros.

Para activar esta funcionalidad será necesario realizar una serie de operaciones tanto a nivel del servidor de aplicaciones (*Oracle WebLogic Server*) como a nivel de *Oracle Service Bus*.

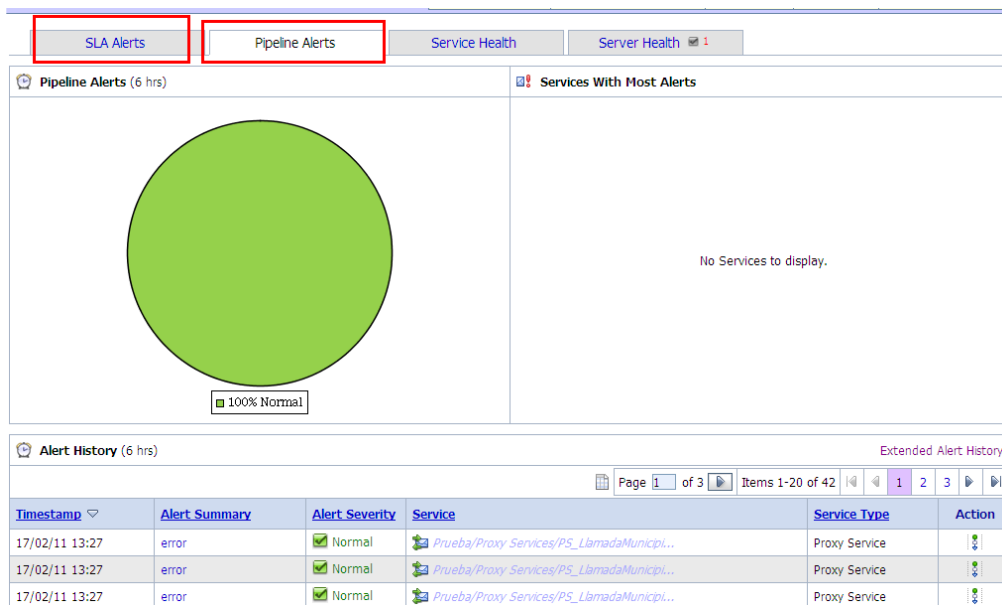
Para ver el detalle de las operaciones a realizar ver⁷ "[Configuración de traps SNMP en OSB 11g](#)".

REPORTING

Permite el envío de las alertas por medio de informes.

ALERT LOGGING

Oracle Service Bus permite el envío de las alertas al panel de control, es decir, si activamos esta opción, las alertas aparecerán en la siguiente pantalla. Si la desactivamos, las alertas no se mostrarán (tanto las alertas SLA como las alertas de error).



E-MAIL

Otra opción es enviar las alertas por correo electrónico. Para ello lo único necesario es añadir el servidor SMTP e indicar la dirección de correo electrónico.

⁷ <http://blogs.oracle.com/osbuniverse/snmpconfigosb/resources/article.pdf>

JMS

Permite enviar las alertas a una cola JMS indicada.

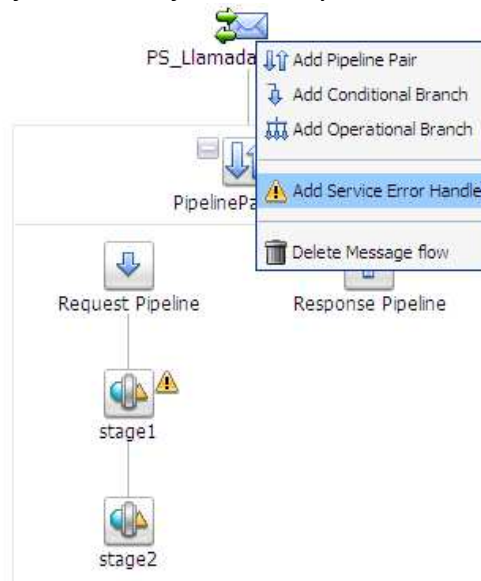
TIPOS DE ALERTAS

Para poder ver las alertas y los informes es necesario activar la monitorización (ya sea en el *Business Service* o en el *Proxy Service*, dependiendo de dónde estemos trabajando).

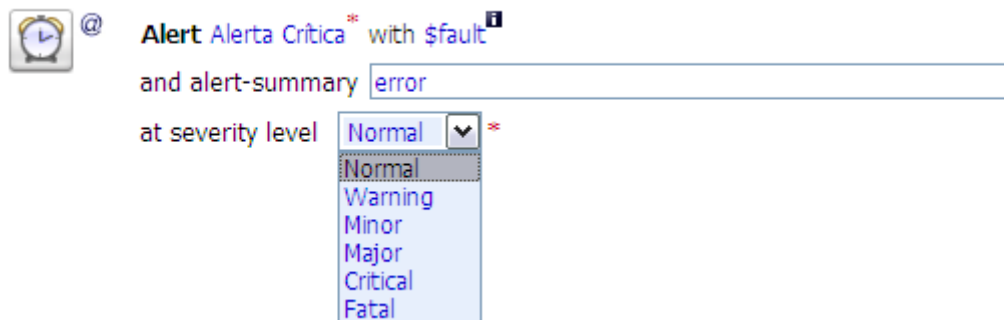
Tanto en el *Proxy Service* como en el *Business Service*, el OSB nos permite indicar qué alertas queremos que se notifiquen y cuáles no, es decir, indicamos a partir de qué nivel de gravedad queremos que se muestren. En el siguiente ejemplo, indicamos que se muestren las alertas con nivel de gravedad a partir de Warning:

ALERTAS DE ERROR

Permite añadir controladores de errores para la detección de fallos. Dichos controladores se pueden añadir a nivel de “pipeline”, “stage” o “Proxy Service”, es decir, se añaden en el flujo de mensajes del *Proxy Service*.



Permite añadir el destino de la alerta, es decir, dónde se va a enviar, qué va a contener la alerta (en este caso añadimos \$fault que contiene el error), una pequeña descripción y el nivel de gravedad de la alerta.



ALERTAS SLA

No sólo sirven para la detección de fallos sino que también nos permite mostrar el éxito de servicios que se están acercando a su límite de rendimiento. Por ejemplo, si el servicio debe ser capaz de responder a una solicitud en menos de 2 segundos, y normalmente responde a estas solicitudes en 0,8 segundos, se deben utilizar múltiples alertas SLA: una con la condición de error (es decir, más de 2 segundos) y otros para ayudar a alertar a su equipo de operaciones cuando el tiempo de servicio empieza a exceder su tiempo normal (es decir, > 1 segundos y, posiblemente, > 1.8 segundos).

El objetivo del SLA no sólo sirve para la notificación de fallos, sino para servir como una herramienta para ayudar a prevenir fallos.

Las reglas de negocio de las alertas SLA se pueden asignar tanto al *Proxy* como al *Business Service*.

Configuration Details
Operational Settings
SLA Alert Rules
Policies
Security

Summary of SLA Alert Rules

Name	SLA State	Severity	Aggr. Interval	Expiration Date	Stop Processing	Frequency	Options
lento2	Enabled	Normal	0 hr(s) 1 mins	None	No	Every Time	aje

Back
Add
Update
Reset

Configuración de Condiciones de Regla de Alerta

Seleccione Intervalo de Agregación para la condición: 0 horas y 10 mins.

Expresión Simple

Recuento de Errores

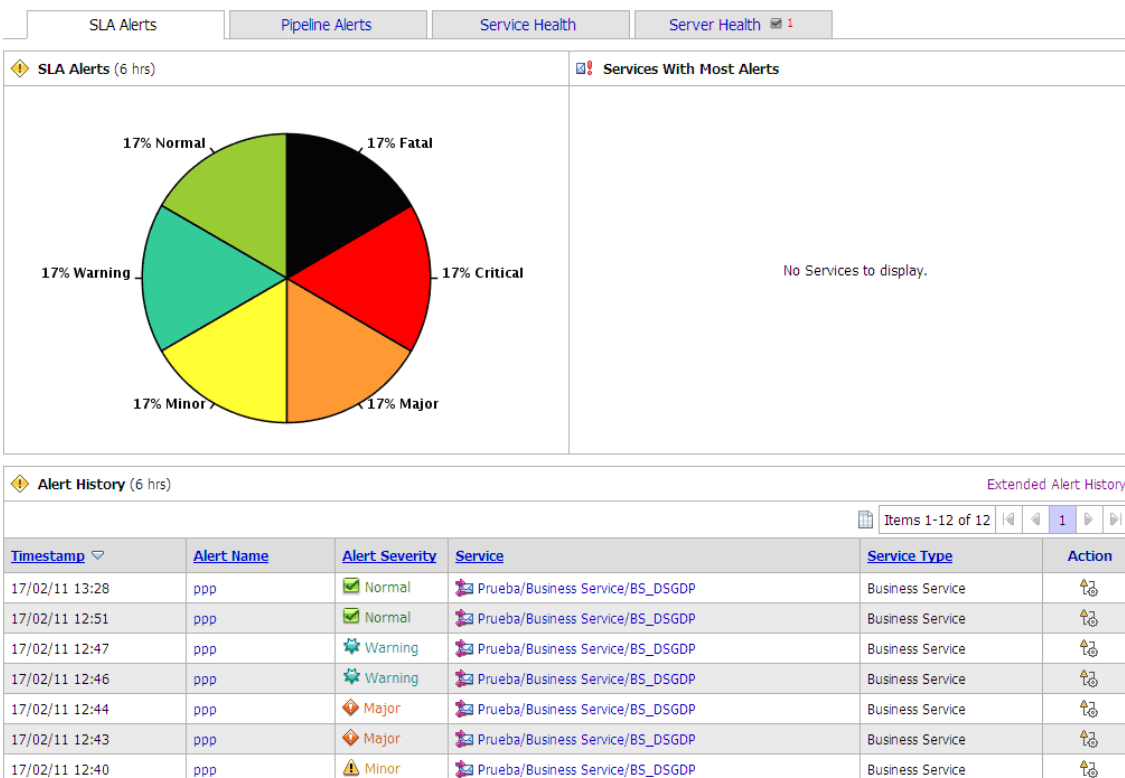
Expresión Compleja

And

<< Anterior

Una vez creadas las alertas, cada vez que hagamos una llamada a dicho Proxy y se cumpla una de las reglas asignadas, saltará la alerta específica. Se mostrará en el panel de control del OSB o según el método de envío que hemos especificado al crear la alerta.

Cada alerta se puede configurar con un nivel de gravedad: Fatal, Critical, Major, Minor, Warning y Normal. Además permite elegir las alertas a mostrar, por ejemplo, mostrar las alertas con nivel de gravedad a partir de Warning.



PRESENTACIÓN DE INFORMES

A parte de las alertas, OSB permite crear informes en tiempo de ejecución.

La presentación de informes se basa en la grabación de partes de mensaje en una base de datos. A continuación, se pueden ver cualquiera de estas entradas del informe en la consola del OSB o acceder a ellas desde una herramienta de información externa.

Para crear un informe se debe utilizar *“Report action”* en el flujo del mensaje del *Proxy Service*.

Para la gestión de los informes se utiliza un proveedor de información. Proporciona el transporte y almacenamiento para informar sobre los eventos.

De forma predeterminada, OSB viene preconfigurado con una presentación de un proveedor de informes JMS. Este proveedor JMS primero encola el registro de información, y a continuación, lo desencola y escribe los registros en la base de datos. Este diseño asíncrono mejora el rendimiento de los módulos de información dentro de OSB. El nombre del servidor JMS es *wlsbjmsserver*, y el nombre del almacén de datos es *wlsbjmsrpDataStore*, que se encuentra en *Weblogic*.

Se puede cambiar el almacén de datos asociado con el proveedor de información JMS (se pueden utilizar Oracle, Microsoft SQL Server o MySQL ya que se tiene los script de bases de datos).

Destinos (Filtrado: Existen Más Columnas)

Producción ▼ Consumo ▼ Inserción ▼

Mostrando 1 a 10 de 15 Anterior | Siguiente

<input type="checkbox"/>	Nombre ↕	Mensajes Actuales	Mensajes Pendientes	Recuento Alto de Mensajes	Mensajes Recibidos	Umbral de Mensajes	Tipo de Destino	Estado	Producción en Pausa	Inserción en Pausa	Consumo en Pausa
<input type="checkbox"/>	jmsResources! ColaErrorLento60segundos	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! PS2LentoPetición656390376	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider.queue	0	0	2	53	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.jmsprovider.error.queue	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wli.reporting.purge.queue	0	0	2	6	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.email	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.file	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.ftp	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! wlsb.internal.transport.task.queue.sftp	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false
<input type="checkbox"/>	jmsResources! WSLentoPuntoAPunto_PS2ColaError	0	0	0	0	0	Queue	advertised_in_cluster_indi	false	false	false

Producción ▼ Consumo ▼ Inserción ▼

Mostrando 1 a 10 de 15 Anterior | Siguiente

También se puede utilizar OSB sin proveedor de información lo único que los datos no se almacenarán en ninguna base de datos.

Se puede utilizar *“Report action”* en el flujo de mensajes de cualquier *Proxy Service*. La información se agrega en una base de datos. OSB proporciona una interfaz de usuario de informes que permite recopilar la información que se necesita con facilidad.

Los informes los podemos ver en la consola del OSB:

ORACLE® Service Bus 11gR1

Bienvenido, weblogic. Conectado a: poc_osbsoa

Inicio | Consola de Oracle WLS | Desconectar | Ayuda | Servicios de Soporte Oracle | Acerca de Serv

Resumen de Informes de Mensajes Depurar

Filtros de Informes de Mensajes

Rango de Fechas: ☐ Todo ☐ Mensaje del D/MM/YY 12 : 00 AM al D/MM/YY 12 : 00 AM ☒ De los últimos 0 días 00 horas 30 minutos

Nombre de Servicio Entrante:

Código de Error:

Índice de Informes: (Separe los distintos índices con comas)

Búsqueda Ver Todo

Índice de Informes	Registro de Hora de Base de Datos	Servicio Entrante	Ruta de Acceso	Código de Error
myID=1;myName=name1	15/10/10 14:07	Reporting	Reporting_S8	

Elementos 1 - 1 de 1

Principio

Operaciones

Explorador de Recursos

Explorador de Proyectos

Configuración de Seguridad

Administración del Sistema

Los informes sólo nos permiten obtener información de los mensajes, por ejemplo:

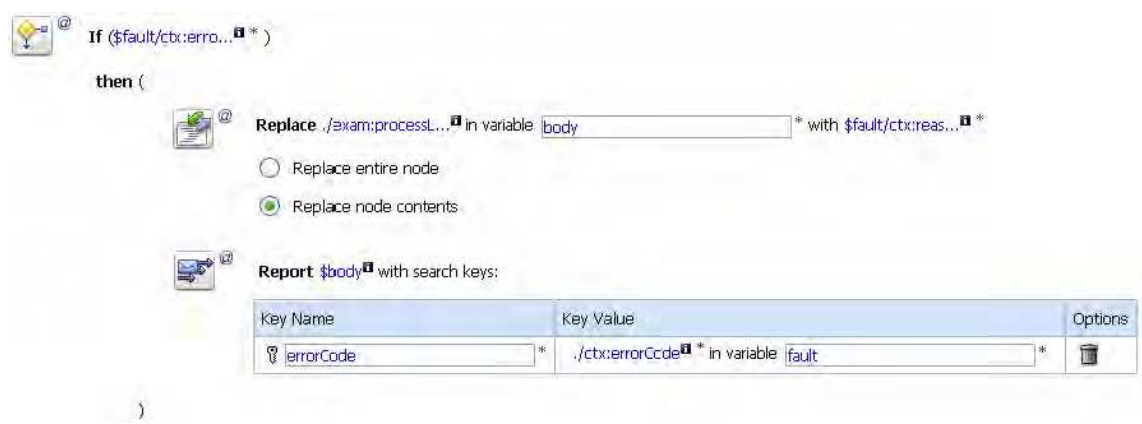
- Mostrar el contenido de la variable prueba en un momento determinado. Nos permite añadir un nombre de clave, es decir, un identificador del informe al cual le asignamos un valor que puede ser un campo de otra variable, por ejemplo.



Informe \$Prueba con claves de búsqueda:

Nombre de la Clave	Valor de Clave	Opciones
<input type="text"/>	<XPath> en variable <input type="text"/>	<input type="text"/>

- En el siguiente ejemplo, si se produce un error cuyo código de error es BEA-382505, nos muestra un informe cuyo nombre de clave es el código de error y nos muestra el contenido de la variable \$body.



Si ejecutamos un servicio el cual genera un informe y la cola JMS falla, el servicio da error y no se ejecuta correctamente.

Diseño proceso de gestión de cambios con *Oracle Service Bus*

Tanto en la consola del OSB como en Eclipse se permite importar / exportar los proyectos y su configuraciones.

La consola del OSB permite guardar como metadatos las configuraciones de los recursos y exportarlos en archivos .JAR para poder ser utilizados en otros dominios de *Oracle Service Bus*.

Además de permitir importar / exportar los recursos, también permite importar / exportar los proyectos enteros en un archivo .JAR. Utilizando las características del sistema de control de código fuente existente junto con los archivos de configuración JAR, ofrece versión y la gestión del cambio para configuraciones de *Oracle Service Bus*.

Otro punto a tener en cuenta es la localización de los recursos. Los recursos no son los mismos en preproducción que en producción, es decir, son los mismos pero no se encuentran en el mismo sitio. Esta localización viene definida en el *Business Service* como URI de punto final. Por todo esto, a la hora de importar un proyecto a un OSB se debe modificar esta URI en los casos que sea necesario.

Crear un Servicio de Negocio (CanalEntregaGarantizada/Business Services/BusinessService1)

Configuración de Transporte

Protocolo*	http				
Algoritmo de Equilibrio de Carga	asignación en rueda (round-robin)				
URI de Punto Final*	<p>Formato: http://host:port/someService</p> <p>http://localhost:7001/CanalEntregaGarantizada/Business_Services/BusinessSe Agregar</p> <table border="1"> <thead> <tr> <th>URI EXISTENTES</th> <th>OPCIONES</th> </tr> </thead> <tbody> <tr> <td>http://.../WSLento60segundos-WSLento60segundos-context-root/WSLento60segundosPort</td> <td>↑ ↓ ✎ 🗑</td> </tr> </tbody> </table>	URI EXISTENTES	OPCIONES	http://.../WSLento60segundos-WSLento60segundos-context-root/WSLento60segundosPort	↑ ↓ ✎ 🗑
URI EXISTENTES	OPCIONES				
http://.../WSLento60segundos-WSLento60segundos-context-root/WSLento60segundosPort	↑ ↓ ✎ 🗑				
Recuento de Reintentos	0				
Intervalo de Iteración de Reintentos	30				
Reintentar tras Errores de Aplicación	<input checked="" type="radio"/> Sí <input type="radio"/> No				

<< Anterior Siguiente >> Último >> Cancelar

Anexo IV: Glosario de términos

Servicio Web o Web Service

Un servicio web (*Web Service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

SOA

Las siglas SOA son la abreviatura de *Service Oriented Architecture*.

Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente, servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOAP

SOAP (siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

WSDL

WSDL son las siglas de *Web Services Description Language*, un formato XML que se utiliza para describir servicios Web. La versión 1.0 fue la primera recomendación por parte del W3C y la versión 1.1 no alcanzó nunca tal estatus. La versión 2.0 se convirtió en la recomendación actual por parte de dicha entidad.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligán después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML *Schema*. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML *Schema*. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

XSLT

Al igual que XML, XSLT (*XML Stylesheets Language for Transformation*, o lenguaje de transformación basado en hojas de estilo) es un lenguaje de programación. Forma parte de la *trilogía transformadora* de XML, compuesta por las CSS (*Cascading Style Sheets*, hojas de estilo en cascada), que permite dar una apariencia en el navegador determinada a cada una de las etiquetas XML.

XSLT es pues, un lenguaje que se usa para convertir documentos XML en otros documentos XML; puede convertir un documento XML que obedezca a un DTD a otro que obedezca otro diferente, un documento XML bien formado a otro que siga un DTD, o, lo más habitual, convertirlo a "formatos finales", tales como WML (usado en los móviles WAP) o XHTML.

Los programas XSLT están escritos en XML, y generalmente, se necesita un procesador de hojas de estilo, o *stylesheet processor* para procesarlas, aplicándolas a un fichero XML.

UDDI

UDDI son las siglas del catálogo de negocios de Internet denominado *Universal Description, Discovery and Integration*. El registro en el catálogo se hace en XML. UDDI es una iniciativa industrial abierta (sufragada por OASIS) entroncada en el contexto de los servicios Web. El registro de un negocio en UDDI tiene tres partes:

- Páginas blancas - dirección, contacto y otros identificadores conocidos.
- Páginas amarillas - categorización industrial basada en taxonomías.
- Páginas verdes - información técnica sobre los servicios que aportan las propias empresas.

UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

JMS

Es el acrónimo de las palabras *Java Messaging Service*.

Java Message Service (JMS) es un estándar de mensajería que permite a los componentes de la aplicación basada en la plataforma *Java 2 Enterprise Edition* (J2EE) crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

El servicio de mensajería instantánea también es conocido como Middleware Orientado a Mensajes (MOM por sus siglas en inglés) y es una herramienta universalmente reconocida para la construcción de aplicaciones empresariales.

JCA

J2EE Connector Architecture (JCA) define una arquitectura estándar para conectar la plataforma J2EE con recursos externos y EIS heterogéneos como por ejemplo RDBMS o ERP. Define un mecanismo escalable y seguro para dar soporte a la integración de sistemas de datos con servidores J2EE y sus aplicaciones.

JCA permite que un proveedor proporcione un adaptador de recursos estándar para su producto el cual es conectado en un servidor de aplicaciones J2EE para agregar la infraestructura necesaria para utilizarlo. Por tanto, un adaptador de recursos es un driver de nivel de sistema que es utilizado por una aplicación Java para conectar con un EIS.

Así, un servidor que da soporte a JCA asegura la conectividad con múltiples sistemas y productos de diferentes vendedores, y un producto que provee de un adaptador de recursos tiene la capacidad de ser utilizado por cualquier servidor que de soporte a JCA.

SCA

SCA es un estándar de OASIS originalmente creado por diferentes vendedores como BEA, IBM, Oracle, SAP, etc. La especificación SCA define como crear un componente y como estos componentes interactúan para formar una aplicación. Los componentes en SCA pueden ser contruidos en Java o en otros lenguajes y además permite interactuar con otras tecnología como JEE, Spring o BPEL. SCA define un mecanismo común de ensamblaje que indica como los componentes son combinados dentro de la aplicación.

WORKFLOW

Decimos *WorkFlow* “Tradicional”, porque la forma actual y bajo SOA de implementar un *workflow* es BPM, pero existen aún tecnologías *WorkFlow* tradicionales (no BPM), pero la mayoría se está transformando en soluciones BPM.

WorkFlow (Tradicional) es la tecnología que permite coordinar actividades humanas (realizadas por personas, *Human Task*), aquí se definen roles, actividades, reglas de negocio, en buenas cuentas un flujo de trabajo (*WorkFlow*), pero que a diferencia de BPM, no contempla en forma natural los “servicios de negocio”, es decir no contempla las actividades de sistemas (*System Task*) bajo el estándar SOA.

En una solución (*WorkFlow* tradicional) el participante (usuario) debe indicar el cumplimiento de la tarea (cambio de estado), para de esta forma continúe el flujo de trabajo, y cuando debe interactuar con algún sistema, debe levantar dicho sistema y realizar la actividad en él.

Esa es otra gran diferencia con BPM, en BPM se trata de proveer una sola interfaz para un participante del proceso, ocultando la interacción con los sistemas, mientras en un *Workflow* (tradicional) la persona debe interactuar con distintos ambientes o aplicaciones, dicho de otra forma: la persona debe manejar distintas aplicaciones (sistemas), y además registrar su avance en el *Workflow*.

Con *Workflow* (al igual que BPM) se le da seguimiento y control a los procesos de negocio, es decir, podemos saber el estado actual de cada proceso, en qué lugar del flujo se encuentra. Otra característica que BPM heredó de los *Workflow*, es que a través del proceso generalmente fluye información (documentos, datos), lo que se llama a metadata, u objeto de negocio (BPM).

ESB

Enterprise Service Bus (ESB) consiste en un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos. Los desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas.

Un ESB generalmente proporciona una capa de abstracción construida sobre una implementación de un sistema de mensajes de empresa que permita a los expertos en integración explotar el valor del envío de mensajes sin tener que escribir código. Al contrario que sucede con la clásica integración de aplicaciones de empresa (*Enterprise Application Integration*, EAI) que se basa en una pila monolítica sobre una arquitectura *hub and spoke*, un bus de servicio de empresa se construye sobre unas funciones base que se dividen en sus partes constituyentes, con una implantación distribuida cuando se hace necesario, de modo que trabajen armoniosamente según la demanda.

BPEL

Business Process Execution Language (BPEL), diminutivo de *Web Services Business Process Execution Language* (WS-BPEL) es un lenguaje estandarizado por OASIS para la composición de servicios web. Está desarrollado a partir de WSFL y XLANG, ambos lenguajes orientados a la descripción de servicios Web. Básicamente, consiste en un lenguaje basado en XML diseñado para el control centralizado de la invocación de diferentes servicios Web, con cierta lógica de negocio añadida que ayuda a la programación en gran escala. Antes de su estandarización se denominaba BPEL4WS.

BPM

Se llama Gestión de procesos de negocio (*Business Process Management* o BPM en inglés) a la metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. Como su nombre sugiere, BPM se enfoca en la administración de los procesos del negocio.

A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos. La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de los mismos. La administración de los procesos permite asegurar que los mismos se ejecuten eficientemente, y la obtención de información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias en los mismos, y actuar sobre las mismas para optimizarlos.

Para soportar esta estrategia es necesario contar con un conjunto de herramientas que den el soporte necesario para cumplir con el ciclo de vida de BPM. Este conjunto de herramientas son llamadas *Business Process Management System* y con ellas se construyen aplicaciones BPM.

Existen diversos motivos que mueven la gestión de procesos de negocio, entre los cuales se encuentran:

- Extensión del programa institucional de calidad.
- Cumplimiento de legislaciones.
- Crear nuevos y mejores procesos.
- Entender qué se está haciendo bien o mal a través de la comprensión de los procesos.
- Documentar procesos para subcontratación y definición del *Service Level Agreement* (SLA).
- Automatización de procesos.
- Crear y mantener la cadena de valor.

BPMN

La notación para gestión de procesos del negocio (BPMN) es el estándar mas reciente para modelado de procesos del negocio y servicios web.

BPMN es una notación necesaria para expresar los procesos de negocio en un único diagrama de proceso de negocio (BPD).

BPMN permite hacer un mejor uso de la gestión de procesos del negocio (BPM), ya que normaliza el método de notación que sirve como ayuda en la automatización de los procesos.

BAM

BAM, *Business Activity Monitoring*, es un método para analizar los procesos de negocio examinando las actividades que se ejecutan en los sistemas IT que soportan estos procesos. BAM permite monitorizar los procesos a partir de pocos datos procesados en vivo, facilitando la toma de decisiones de corto plazo en vez de la planeación de largo plazo posible con otras herramientas de inteligencia de negocio. En BAM, se configura un programa cuyo único propósito es “estudiar” una función específica de IT, o conjunto de funciones, que soportan un proceso de negocio específico. De la

información tomada en el proceso, los gerentes del negocio pueden adquirir una visión de las operaciones y tomar decisiones de acuerdo a esta visión.

Prueba de Concepto OSB

Almudena Ochoa Lainez

INDICE

◉ INTRODUCCIÓN

- › ¿Qué es una prueba de concepto?
- › Solicitantes de la prueba.
- › Objetivos del proyecto

◉ CONCEPTOS

- › ¿Qué es OSB?
- › Herramientas necesarias
- › Conceptos generales
- › Casos de prueba
- › Ejemplo

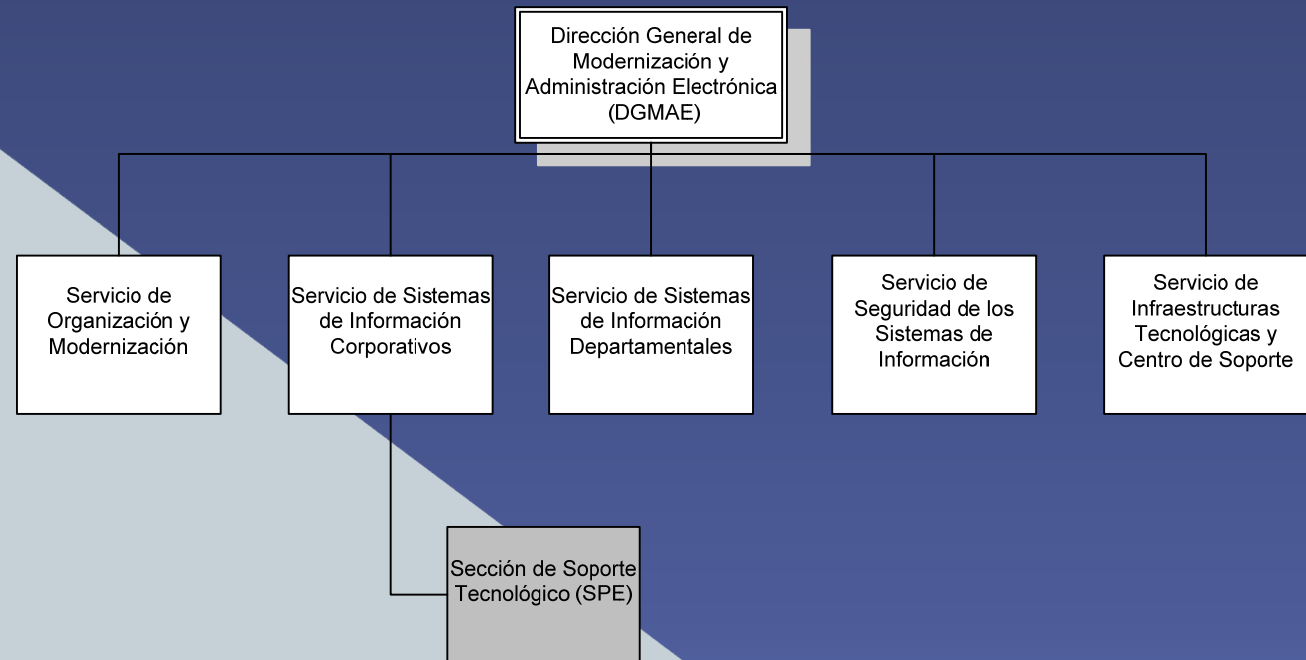
◉ Conclusiones

INTRODUCCIÓN

¿Qué es una prueba de concepto?

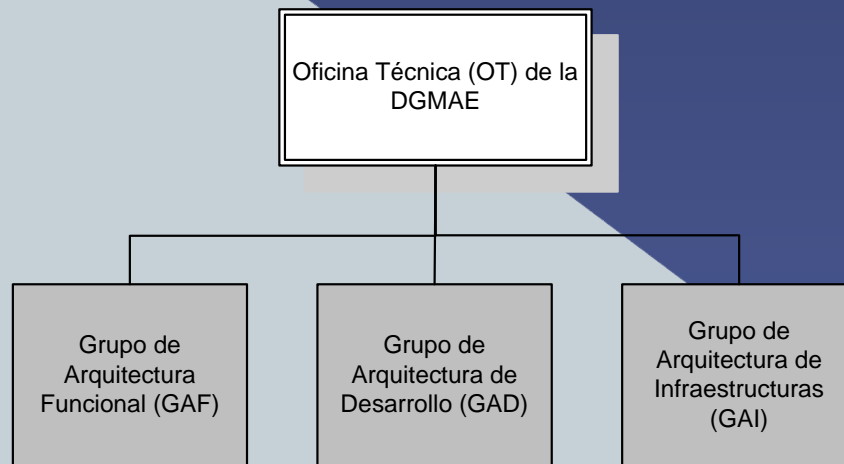
- Consiste en un conjunto de casos propuestos por los solicitantes de la prueba de concepto, describiendo situaciones y características que consideran de interés.
- Como resultado se genera un informe de resultados de cada una de las pruebas realizadas, que ayuda a los solicitantes a tomar decisiones.

Solicitantes de la prueba



Objetivo SPE: Gestión de todos los estándares tecnológicos de información y aplicaciones de la administración.

Solicitantes de la prueba



Objetivo OT: proporcionar ayuda estratégica a la dirección, implantar una cultura de reutilización y facilitar la visión del escenario tecnológico a los equipos de proyecto.

Objetivo GAI: vela por la estandarización de la arquitectura de Ejecución. Determina el escenario tecnológico.

Objetivo GAD: vela por la estandarización de las herramientas y plataformas de desarrollo.

Objetivo GAF: vela por la estandarización de los componentes software reutilizables.

Objetivos del proyecto

- Conocer, mediante la realización de la prueba de concepto, las características, requerimientos, ventajas y limitaciones de un bus de integración en los futuros proyectos de Gobierno de Navarra.
- Obtener un informe de resultados de los diferentes casos de uso planteados que permita a los solicitantes conocer el resultado de la prueba y permita la futura toma de decisiones.

CONCEPTOS

¿Qué es OSB?

- Pertenece a la arquitectura SOA.
- Es uno de los componentes del producto SOA SUITE.
- OSB es un intermediario que procesa los mensajes entrantes de solicitud de servicio, determina la lógica de enrutamiento, y transforma estos mensajes para la compatibilidad con los consumidores de servicios.

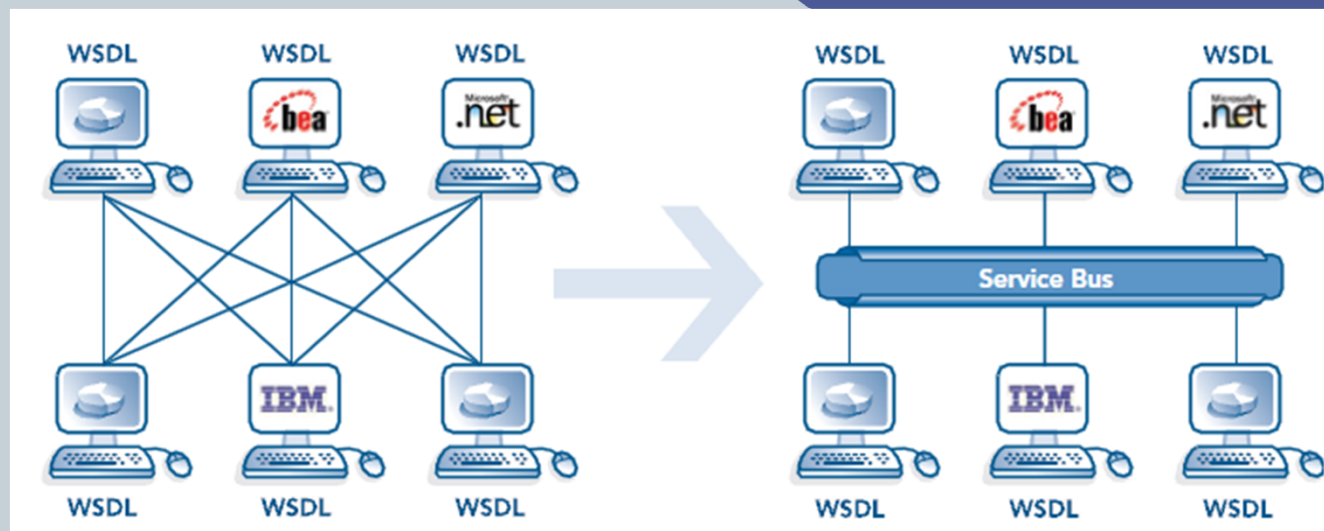
¿Qué es OSB?

⦿ Principales características:

- › Transparencia de las ubicaciones
- › Transparencia del transporte
- › Soporte multiprotocolo
- › Calidad de servicio
- › Patrones de intercambio de mensajes
- › Enrutamiento basado en el contenido
- › Transformación

¿Qué es OSB?

- A continuación se muestra el problema de la explosión de conexiones que generan los enfoques de integración punto a punto.
- Usando *Oracle Service Bus*, cada aplicación se conecta sólo una vez a una infraestructura troncal común.

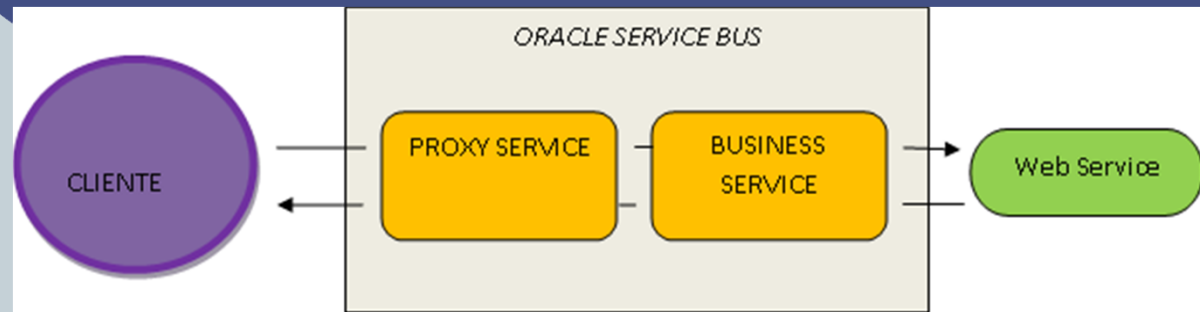


Herramientas necesarias

- ⦿ Herramientas necesarias para un desarrollador:
 - > Eclipse 3.5.2 + OEPE11g R1
 - > OSB Console (11.1.1.3.0)
 - > JDeveloper 11.1.1.3.0
 - > OSB en local
- ⦿ Otras herramientas utilizadas en el desarrollo de este proyecto:
 - > Visual Studio 2008
 - > NetBeans 6.9.1

Conceptos generales

Construido sobre la tecnología *Oracle WebLogic Server*.

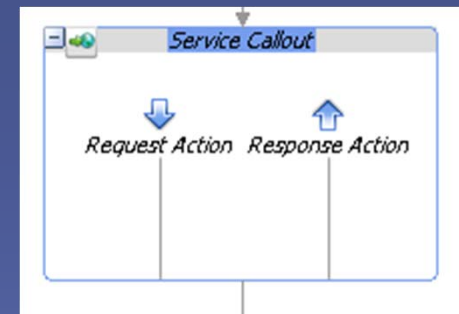
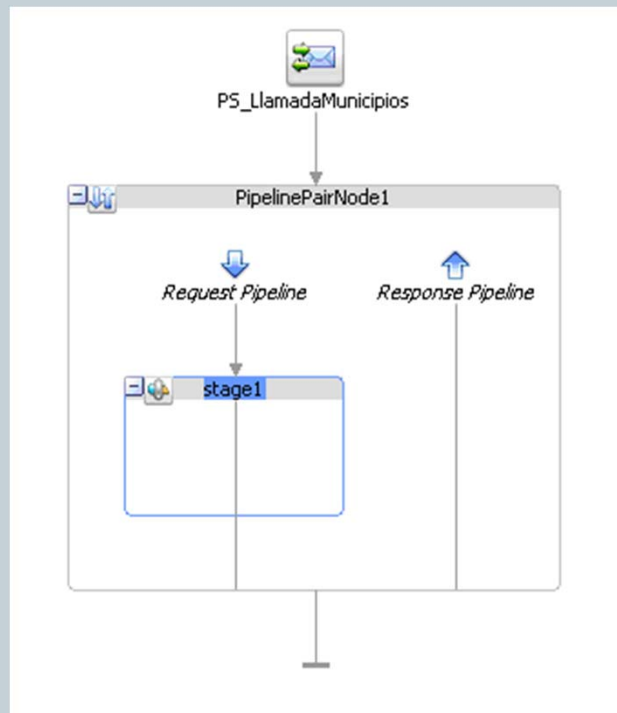


Business Service es la definición de un servicio de empresa que intercambia mensajes durante los procesos de negocio.

Proxy Service es un intermediario de un *Web Service* que *Oracle Service Bus* implementa a nivel local en el servidor *WebLogic*.
Flujo de mensajes es la implementación de un *Proxy Service*.

Conceptos generales

Un *pipeline* es una ruta de procesamiento dentro del flujo de mensajes.



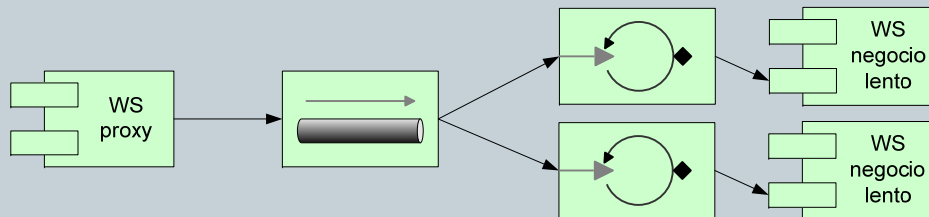
The **Service Callout** configuration window shows the following settings:

- Service:** Prueba/Business Service/BS_DSGBP (highlighted with a red box)
- Invoking:** mostrarDatos (highlighted with a red box)
- Request Variable:** body (highlighted with a red box)
- Response Variable:** Salida (highlighted with a red box)

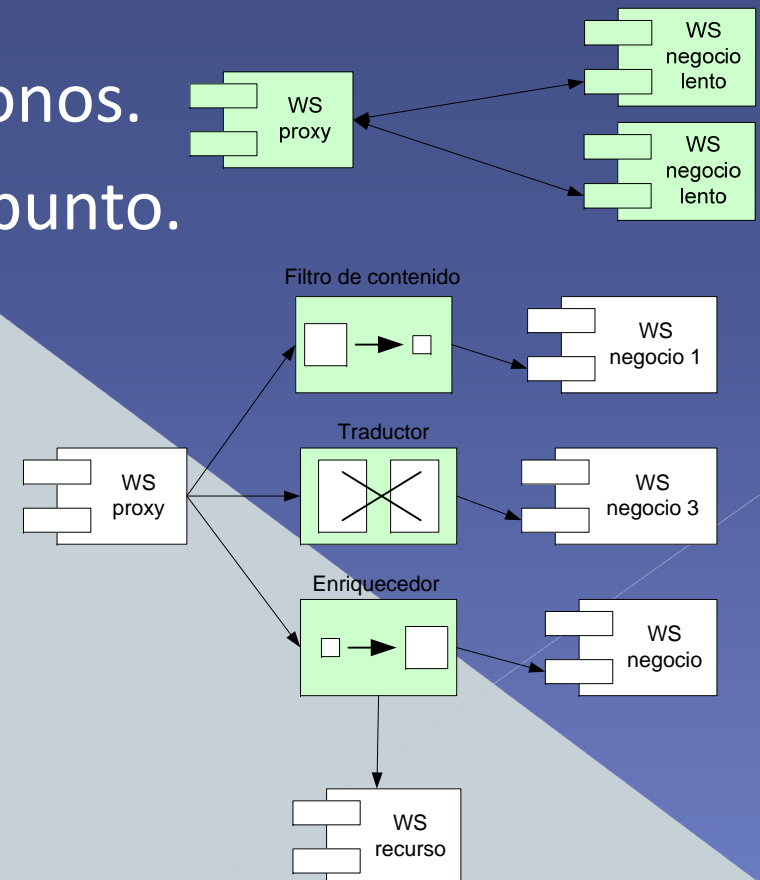
Other options include **Configure Soap Body** and **Configure Payload Document** (selected).

Casos de prueba

- Publicar y proteger con *WS-Security* un WS sin protección.
- Escalabilidad en servicios síncronos.
- Escalabilidad en canal punto a punto.

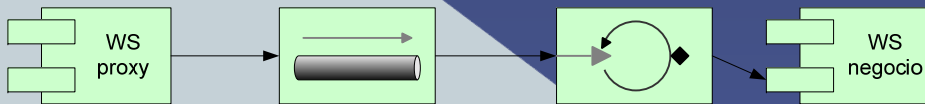


- Transformación de mensajes.

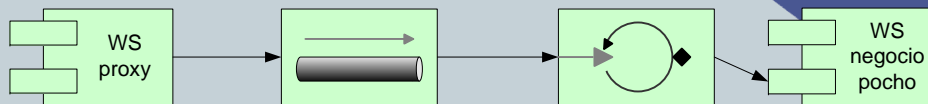


Casos de prueba

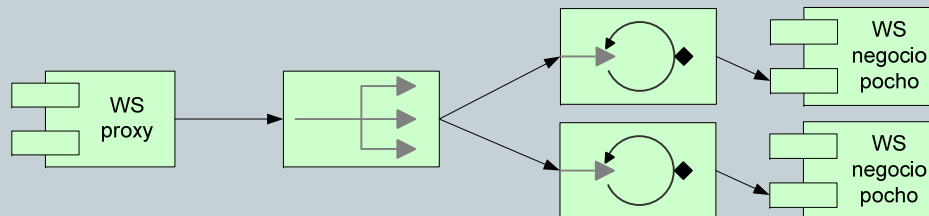
- ◉ Consumidor de mensajes.



- ◉ Canal punto a punto confiable.

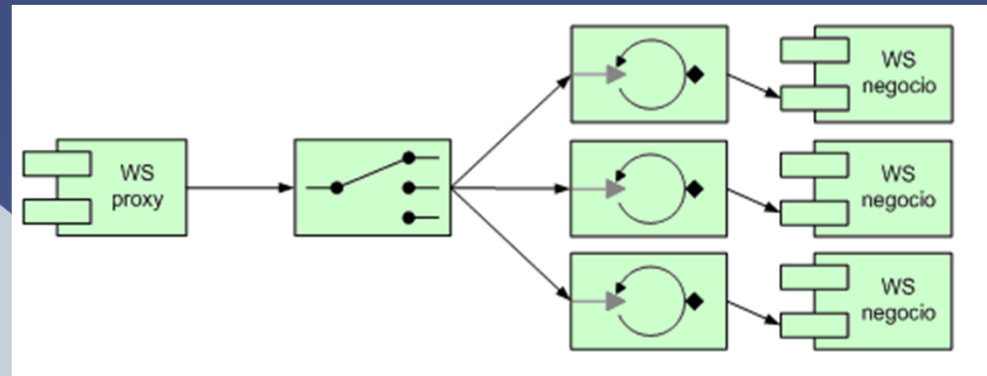


- ◉ Canal de publicación / subscripción.



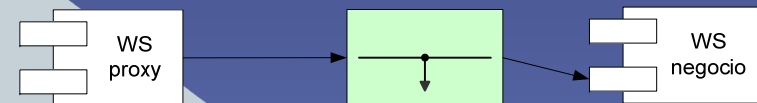
Casos de prueba

- Encaminamiento basado en el contenido.

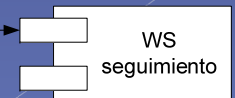
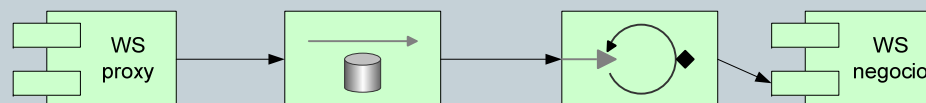


- Separador.

- Escucha en llamada síncrona.

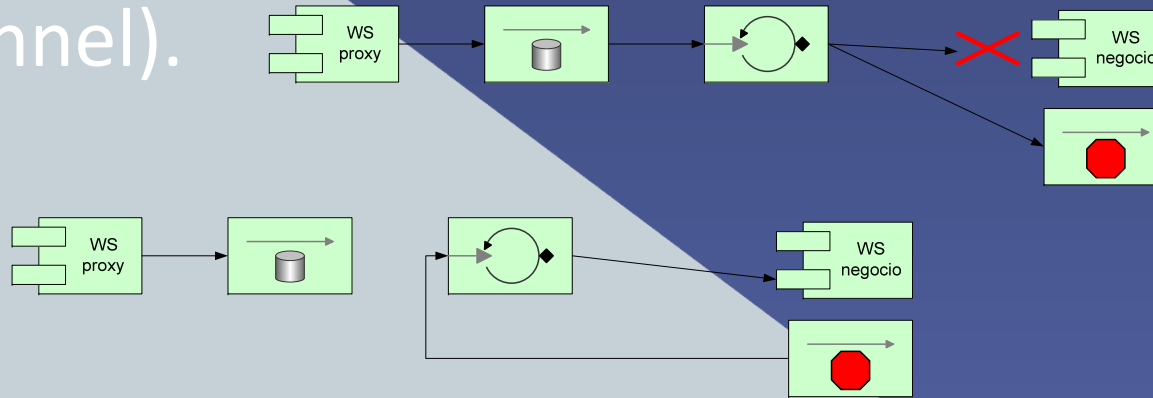


- Canal de entrega garantizada.



Casos de prueba

- Canal de mensajes fallidos (dead letter channel).



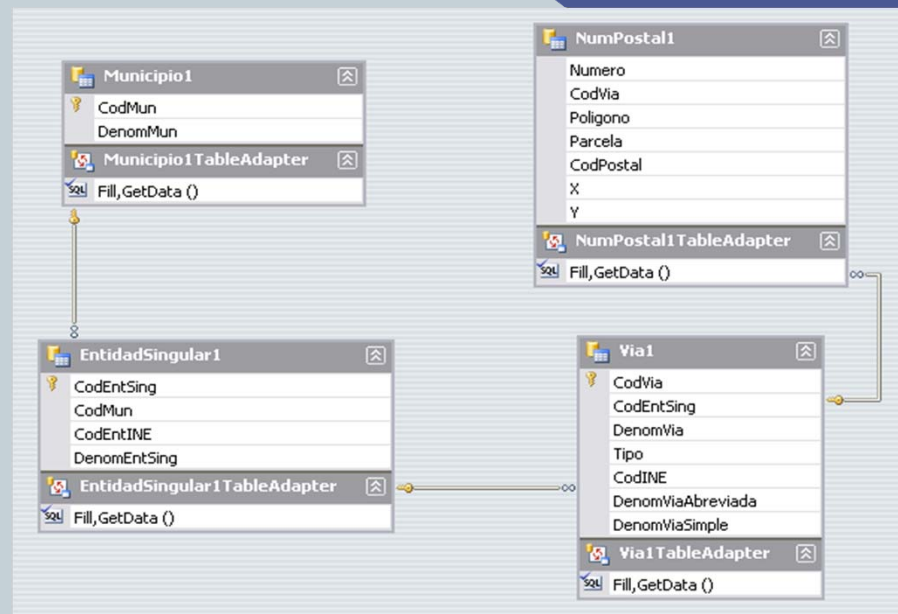
- Invocación de WS con tipos de datos propietarios de .Net desde Java.

Entregables de diseño

- ⦿ Informes con *Oracle Service Bus*.
- ⦿ Diseño proceso de gestión de cambios con *Oracle Service Bus*.

Ejemplo

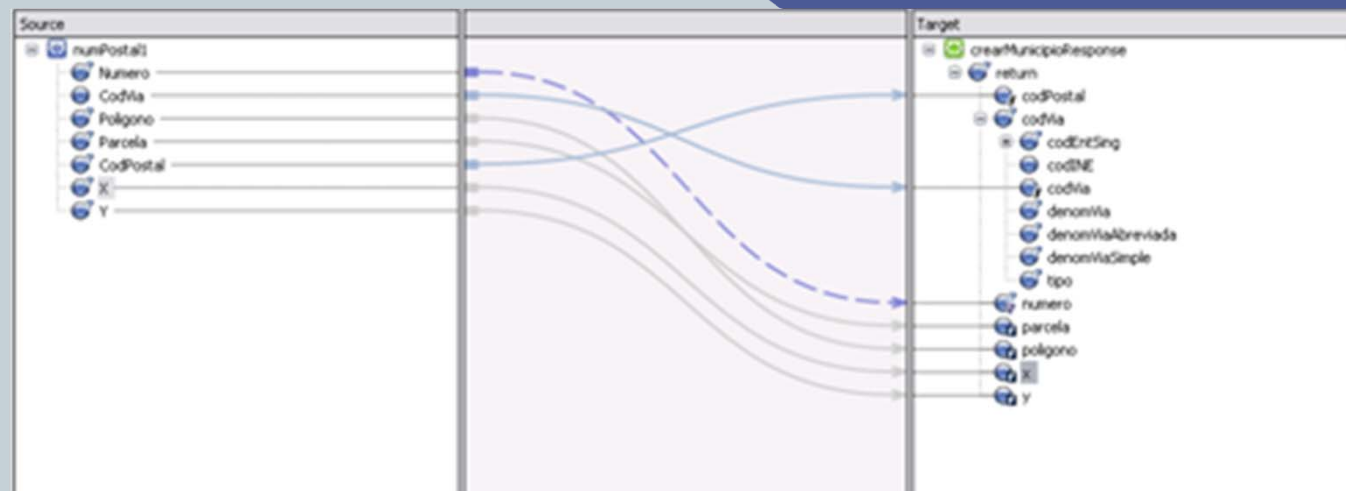
- Caso de Uso: Invocación de WS con tipos de datos propietarios de .Net desde Java.
 - Web Services .Net con tipo de datos DataSet.
 - No interoperable con Java.



Ejemplo

● Solución:

- Implementar un WS en Java con la misma estructura que el DataSet de .Net.
- Desde el Proxy Service realizar la transformación del tipo de dato que devuelve el servicio de .Net a lo que espera el WS de Java. Transformación XQuery.



CONCLUSIONES

Conclusiones

- De los 13 casos realizados los 13 se han desarrollado con éxito.
- El OSB:
 - Permite proteger con WS-Security cualquier servicio. Java es interoperable con cualquier política. En .Net 3.5 es necesario que las políticas OWSM sigan el estándar *WS-Security* 1.1.
 - Tanto Escalabilidad síncrona como punto a punto: no se reduce a la mitad el tiempo en procesar las llamadas pero disminuye considerablemente el tiempo de ejecución.

	1 URL	2 URLs
9 Llamadas	30 segundos	30 segundos
12 Llamadas	60 segundos	30 segundos
21 Llamadas	90 segundos	60 segundos
30 Llamadas	120 segundos	90 segundos

Conclusiones

- OSB proporciona apoyo a la transformación de mensajes usando dos tecnologías estándares: XSLT y XQuery.
- No es necesario crear un consumidor de mensajes para ser intermediario entre la cola de mensajes y el servicio de negocio. En OSB es la propia cola JMS la que actúa como consumidor de los mensajes ya que dispone de un MDB propio implementado.
- Canal punto a punto confiable: Si se produce un error, el mensaje no se pierde sino que se guarda en una cola JMS.

Conclusiones

- Canal Publicador/subscriber: permite publicar un mensaje en varios subscriptores además de añadir confiabilidad al sistema.
- Split Join: permite dividir una carga útil de servicio, como por ejemplo una orden, en mensajes individuales para el procesamiento concurrente. El procesamiento concurrente, en comparación con el tratamiento secuencial, mejora notablemente el rendimiento del servicio.
- Permite realizar operaciones totalmente transparentes para la aplicación cliente y el servicio de negocio.
- Si en un canal punto a punto se produce un error, por ejemplo se cae el servidor de aplicaciones, los mensajes no se pierden.

Conclusiones

- Canal de mensajes fallidos: Nos permite mover los mensajes de la cola de error a la cola de peticiones para volver a procesarlos una vez que se ha resuelto el problema del *Web Service*.
- Permite crear alertas de error, alertas SLA e informes. Estas alertas pueden ser enviadas por SNMP, a un informe, por correo electrónico y a una cola JMS.

